PC-8001 mkII

解析マニュアルI

川村 清 著

SHUWA SYSTEM TRADING CO.,LTD.

PC-8001mkII
ANALYTICAL MANUAL

SHUWA SYSTEM TRADING CO.,LTD.

まえがき

パーソナル・コンピュータＰＣ－８００１ｍｋⅡは，我国におけるパーソナル・コンピュータ・ブームの端緒を開くのに多大な役割を演じ多くの実績を誇るＰＣ－８００１の思想を受け継いだ上で，さらに多くの機能強化がなされています．

ＰＣ－８００１と比較したＰＣ－８００１ｍｋⅡの主な特徴は，Ｎ－ＢＡＳＩＣの機能を拡張したN80－ＢＡＳＩＣが採用されたこと，本体内に６４Ｋバイトものメイン ＲＡＭおよび１６Ｋバイトのグラフィクス ＲＡＭが標準実装されたこと，これにより高密度なグラフィクス表現さらには日本語処理が可能になったこと，などがあげられます．

本書は，先に述べた特徴を充分に踏まえた上で，各々のユーザがＰＣ－８００１ｍｋⅡをＰＣ－８００１の後継機種として，また新しい１つのパーソナル・コンピュータとしても思うままに活用できるよう，著者らが独自に解析したＰＣ－８００１ｍｋⅡの内容をまとめあげ，ここに発表するものであります．

読者諸兄におかれては，本書を熟読されることにより，ＰＣ－８００１ｍｋⅡの内部構造を一層深く理解され，高度な活用をされるためのお役に立たんことを深く希望するものであります．

１９８３年１０月
ＦＯＲＥＳＩＧＨＴ企画部代表
川村　清

第 I 巻

第　II　巻

第1章

N-BASICシステム・サブルーチン

N－BASICシステム・サブルーチン

本章は，多くの有用なシステム・サブルーチンおよび主要なシステム・ルーチンの中から，機械語によるＰＣ－８００１ｍｋⅡの活用に不可欠だと思われるものを選択し資料集としてまとめたものです．

なお，本章にまとめた解析内容だけでは間に合わず，より高度な応用を考えておられる場合には，御自分の手によるＢＡＳＩＣの解析をお勧めいたします．Ｎ－ＢＡＳＩＣの明細につきましては，拙著『ＰＣ－８００１　ＢＡＳＩＣ　ＳＯＵＲＣＥ　ＰＲＯＧＲＡＭ　ＬＩＳＴＩＮＧＳ』(秀和システムトレーディング株式会社発行）を参照していただきたいと思います．かならずや，ＢＡＳＩＣ解析のお役に立つことでしょう．

本章の各ルーチンは，エントリ・アドレスによってソーティングし，分類いたしました．

アドレス　各システム・ルーチンのエントリ・ポイントを示します．

機　能　各システム・ルーチンの実行内容を簡単に示します．

レジスタ　各システム・ルーチンの実行後，内容が変更されるレジスタを示します．したがってこれらのレジスタの内容を保存しなければならない場合には，ユーザ・プログラム中でのレジスタ退避が必要です．

解　説　各システム・ルーチンの詳しい使用方法，入出力パラメータなどを説明した上で，場合によっては関連した予備知識を示します．

サンプル　各システム・サブルーチンの使用例として，できる限り多くのサンプル・プログラムを取り上げました．

これらのプログラムは，一部を除いてザイログ形式のＺ８０ニーモニックによる記述を行いアセンブル・リストを掲載しました．あわせてサンプル・プログラムの実行例も示しましたので，各システム・ルーチンの仕様を理解される上でお役立てください．

本章の執筆にあたっては，つぎに示す構成のシステムを使用しております．

1）ＰＣ－８００１ｍｋⅡ　シリアル・ナンバ　３５０２４５８ＣＣ
2）ＰＣ－８８２２　１６ピン・ドット・マトリクス漢字プリンタ
3）ＰＣ－８０３１－２Ｗ　両面倍密度デュアル・ミニ・ディスク・ユニット
4）ＰＣ－８０４６　９インチ・グリーン・ディスプレイ

０ ０ ０ ０ Ｈ　コールド・スタート

アドレス　０ ０ ０ ０ Ｈ

機　　能　コールド・スタート

レジスタ　ＡＦＢＣＤＥＨＬ

解　　説　ＳＴＯＰキーの入力中であれば，００６ＡＨ番地のホット・スタートにジャンプし，ＳＴＯＰキーの入力中でなければ，１７５７Ｈ番地にジャンプすることによりシステムのイニシアライズを行います．

システムのイニシアライズ時には，ＢＡＳＩＣのテキストを消去しますが，００６ＡＨ番地のホット・スタートでは，ＢＡＳＩＣのテキストを保証します．

サンプル

```
                ;
                ; --- cold start or not ---
                ;
                        ORG     0C100H
                ;
C100 2151C1     START:  LD      HL,PROMPT
C103 CD48C1             CALL    DSPMSG
                ;
C106 CD750F     LOOP:   CALL    0F75H           ; input a character from keyboard
C109 CBAF               RES     5,A             ; convert into capital letter
C10B FE59               CP      'Y'
C10D 2806               JR      Z,YES
C10F FE4E               CP      'N'
C111 2820               JR      Z,NOT
C113 18F1               JR      LOOP
                ;
C115 216CC1     YES:    LD      HL,MSGY         ;
C118 CD48C1             CALL    DSPMSG          ;
C11B 060A               LD      B,10            ;
C11D 3E3E       YES1:   LD      A,'>'           ;
C11F CD5702             CALL    0257H           ;
C122 CD3CC1             CALL    WAIT            ; demonstration before cold start
C125 10F6               DJNZ    YES1            ;
C127 2184C1             LD      HL,MSGJ         ;
C12A CD48C1             CALL    DSPMSG          ;
C12D CD3CC1             CALL    WAIT            ;
                ;
C130 C30000             JP      0000H           ;*just cold start
                ;
C133 218DC1     NOT:    LD      HL,MSGN         ;
C136 CD48C1             CALL    DSPMSG          ; jump to system monitor
C139 C3665C             JP      5C66H           ;
                ;
                ; wait subroutine
                ;
                ;    destroys : flags only
                ;
C13C C5         WAIT:   PUSH    BC
C13D 0EFF               LD      C,0FFH
C13F 06FF       WAIT1:  LD      B,0FFH
C141 10FE       WAIT2:  DJNZ    WAIT2
C143 0D                 DEC     C
C144 20F9               JR      NZ,WAIT1
C146 C1                 POP     BC
C147 C9                 RET
                ;
                ; display message subroutine
```

```
                ;
                ;    inputs   : top address of message in register hl
                ;    destroys : register a,f,h,l
                ;
C148 7E         DSPMSG:LD   A,(HL)
C149 A7                AND  A
C14A C8                RET  Z
C14B CD5702            CALL 0257H          ; display a character
C14E 23                INC  HL
C14F 18F7              JR   DSPMSG
                ;
                ; message area
                ;
C151 636F6C64 PROMPT:DEFB 'cold start or not (y/n) ? ',0
C155 20737461
C159 7274206F
C15D 72206E6F
C161 74202879
C165 2F6E2920
C169 3F2000
C16C 7965733A MSGY:  DEFB 'yes:execute cold start ',0
C170 65786563
C174 75746520
C178 636F6C64
C17C 20737461
C180 72742000
C184 206A7573 MSGJ:  DEFB ' just !!',0
C188 74202121
C18C 00
C18D 6E6F3A6A MSGN:  DEFB 'no:jump to monitor !!',0
C191 756D7020
C195 746F206D
C199 6F6E6974
C19D 6F722021
C1A1 2100
                ;
C1A3                   END


*GC100
cold start or not (y/n) ? no:jump to monitor !!
*GC100
cold start or not (y/n) ? yes:execute cold start >>>>>>>>>> just !!
```

０００８Ｈ　ホット・スタート

アドレス	０００８Ｈ
機　　能	ホット・スタート
レジスタ	ＡＦＢＣＤＥＨＬ

解　　説　００６ＡＨ番地へのジャンプを行います．

すなわち，ＣＲＴスクリーン，タイマ等，インタフェース関係のイニシアライズを実行後，インタプリタの制御下に入ります．

サンプル

```
                ;
                ; --- cold start or hot start ---
                ;
                        ORG   0C100H
                ;
C100 215EC1     START:  LD    HL,PROMPT
C103 CD55C1             CALL  DSPMSG
                ;
C106 CD750F     LOOP:   CALL  0F75H          ; input a character from keyboard
C109 CBAF               RES   5,A            ; convert into capital letter
C10B FE43               CP    'C'
C10D 280A               JR    Z,COLD
C10F FE48               CP    'H'
C111 2824               JR    Z,HOT
C113 FE4E               CP    'N'
C115 2829               JR    Z,NOT
C117 18ED               JR    LOOP
                ;
C119 217BC1     COLD:   LD    HL,MSGC        ;
C11C CD55C1             CALL  DSPMSG         ;
C11F 060A               LD    B,10           ;
C121 3E3E       COLD1:  LD    A,'>'          ;
C123 CD5702             CALL  0257H          ;
C126 CD49C1             CALL  WAIT           ; demonstration before cold start
C129 10F6               DJNZ  COLD1          ;
C12B 2191C1             LD    HL,MSGJ        ;
C12E CD55C1             CALL  DSPMSG         ;
C131 CD49C1             CALL  WAIT           ;
                ;
C134 C30000             JP    0000H          ; just cold start
                ;
C137 219AC1     HOT:    LD    HL,MSGH
C13A CD55C1             CALL  DSPMSG
C13D C30800             JP    0008H          ;*hot start


C140 21B1C1     NOT:    LD    HL,MSGN
C143 CD55C1             CALL  DSPMSG
C146 C3665C             JP    5C66H          ; system monitor
                ;
                ; wait subroutine
                ;
                ;    destroys : flags only
                ;
C149 C5         WAIT:   PUSH  BC
C14A 0EFF               LD    C,0FFH
C14C 06FF       WAIT1:  LD    B,0FFH
C14E 10FE       WAIT2:  DJNZ  WAIT2
C150 0D                 DEC   C
C151 20F9               JR    NZ,WAIT1
C153 C1                 POP   BC
```

```
C154 C9                RET
                ;
                ; display message subroutine
                ;
                ;    inputs   : top address of message in register hl
                ;    destroys : register a,f,h,l
                ;
C155 7E         DSPMSG:LD   A,(HL)
C156 A7                AND  A
C157 C8                RET  Z
C158 CD5702            CALL 0257H          ; display a character
C15B 23                INC  HL
C15C 18F7              JR   DSPMSG
                ;
                ; message area
                ;
C15E 636F6C64 PROMPT:DEFB 'cold or hot start (c/h/n) ? ',0
C162 206F7220
C166 686F7420
C16A 73746172
C16E 74202863
C172 2F682F6E
C176 29203F20
C17A 00
C17B 633A6578 MSGC:  DEFB 'c:execute cold start ',0
C17F 65637574
C183 6520636F
C187 6C642073
C18B 74617274
C18F 2000
C191 206A7573 MSGJ:  DEFB '  just !!',0
C195 74202121
C199 00
C19A 683A6578 MSGH:  DEFB 'h:execute hot start !!',0
C19E 65637574
C1A2 6520686F
C1A6 74207374
C1AA 61727420
C1AE 212100
C1B1 6E3A646F MSGN:  DEFB 'n:do nothing !!',0
C1B5 206E6F74
C1B9 68696E67
C1BD 20212100
                ;
C1C1                   END
```

```
*GC100
cold or hot start (c/h/n) ? n:do nothing !!
*GC100
cold or hot start (c/h/n) ? h:execute hot start !!
Ok
mon
*GC100
cold or hot start (c/h/n) ? c:execute cold start >>>>>>>>>> just !!
```

００１３Ｈ　ホット・スタート

アドレス	００１３Ｈ
機　　能	ホット・スタート
レジスタ	ＡＦＢＣＤＥＨＬ

解　　説　００６ＡＨ番地へのジャンプを行います．

すなわち，ＣＲＴスクリーン，タイマ等，インタフェース関係のイニシアライズを実行後，インタプリタの制御下に入りますが，ＢＡＳＩＣのテキストは保証します．

サンプル

```
                ;
                ; --- cold start or hot start ---
                ;
                        ORG   0C100H
                ;
C100 215EC1     START:  LD    HL,PROMPT
C103 CD55C1             CALL  DSPMSG
                ;
C106 CD750F     LOOP:   CALL  0F75H             ; input a character from keyboard
C109 CBAF               RES   5,A               ; convert into capital letter
C10B FE43               CP    'C'
C10D 280A               JR    Z,COLD
C10F FE48               CP    'H'
C111 2824               JR    Z,HOT
C113 FE4E               CP    'N'
C115 2829               JR    Z,NOT
C117 18ED               JR    LOOP
                ;
C119 217BC1     COLD:   LD    HL,MSGC           ;
C11C CD55C1             CALL  DSPMSG            ;
C11F 060A               LD    B,10              ;
C121 3E3E       COLD1:  LD    A,'>'             ;
C123 CD5702             CALL  0257H             ;
C126 CD49C1             CALL  WAIT              ; demonstration before cold start
C129 10F6               DJNZ  COLD1             ;
C12B 2191C1             LD    HL,MSGJ           ;
C12E CD55C1             CALL  DSPMSG            ;
C131 CD49C1             CALL  WAIT              ;
                ;
C134 C30000             JP    0000H             ; just cold start
                ;
C137 219AC1     HOT:    LD    HL,MSGH
C13A CD55C1             CALL  DSPMSG
C13D C31300             JP    0013H             ;*hot start


C140 21B1C1     NOT:    LD    HL,MSGN
C143 CD55C1             CALL  DSPMSG
C146 C3665C             JP    5C66H             ; system monitor
                ;
                ; wait subroutine
                ;
                ;    destroys : flags only
                ;
C149 C5         WAIT:   PUSH  BC
C14A 0EFF               LD    C,0FFH
C14C 06FF       WAIT1:  LD    B,0FFH
C14E 10FE       WAIT2:  DJNZ  WAIT2
```

```
C150 0D               DEC  C
C151 20F9             JR   NZ,WAIT1
C153 C1               POP  BC
C154 C9               RET
              ;
              ; display message subroutine
              ;
              ;    inputs   : top address of message in register hl
              ;    destroys : register a,f,h,l
              ;
C155 7E       DSPMSG:LD   A,(HL)
C156 A7               AND  A
C157 C8               RET  Z
C158 CD5702           CALL 0257H          ; display a character
C15B 23               INC  HL
C15C 18F7             JR   DSPMSG
              ;
              ; message area
              ;
C15E 636F6C64 PROMPT:DEFB 'cold or hot start (c/h/n) ? ',0
C162 206F7220
C166 686F7420
C16A 73746172
C16E 74202863
C172 2F682F6E
C176 29203F20
C17A 00
C17B 633A6578 MSGC:  DEFB 'c:execute cold start ',0
C17F 65637574
C183 6520636F
C187 6C642073
C18B 74617274
C18F 2000
C191 206A7573 MSGJ:  DEFB ' just !!',0
C195 74202121
C199 00
C19A 683A6578 MSGH:  DEFB 'h:execute hot start !!',0
C19E 65637574
C1A2 6520686F
C1A6 74207374
C1AA 61727420
C1AE 212100
C1B1 6E3A646F MSGN:  DEFB 'n:do nothing !!',0
C1B5 206E6F74
C1B9 68696E67
C1BD 20212100
              ;
C1C1                  END
```

```
*GC100
cold or hot start (c/h/n) ? n:do nothing !!
*GC100
cold or hot start (c/h/n) ? h:execute hot start !!
Ok
mon
*GC100
cold or hot start (c/h/n) ? c:execute cold start >>>>>>>>>> just !!
```

００１８Ｈ　デバイスへの１バイト出力

アドレス	００１８Ｈ
機　能	デバイスへの１バイト出力
レジスタ	−−−−−−−−−

解　説　４０Ａ６Ｈ番地へのジャンプを行います．

ＥＢ４９Ｈ番地に格納されているデータが００Ｈの場合にはＣＲＴスクリーンへ，データが０１Ｈ〜７ＦＨの場合にはプリンタへ，データが８０Ｈ〜ＦＦＨの場合にはＣＭＴインタフェースへ，それぞれアキュムレータのデータ１バイトを出力します．

ただし，ＣＭＴインタフェースへの出力を行う場合には，あらかじめインタフェース関係のイニシアライズが行われていることが必要です．

通常は，ＥＢ４９Ｈ番地に００Ｈを与え，ＣＲＴスクリーンへの１文字出力ルーチンとして使用します．

サンプル

```
                ;
                ; --- display or print out characters ---
                ;
                        ORG   0C100H
                ;
C100 AF         START:  XOR   A
C101 3249EB             LD    (0EB49H),A        ; select crt for output
                ;
C104 2162C1             LD    HL,PROMPT
C107 CD59C1             CALL  DSPMSG
C10A CD750F             CALL  0F75H             ; input a character from keyboard
C10D FE03               CP    03H
C10F CA665C             JP    Z,5C66H           ; system monitor if stop key
                ;
C112 F5                 PUSH  AF                ;
C113 CD5702             CALL  0257H             ;
C116 3E0D               LD    A,0DH             ;
C118 CD5702             CALL  0257H             ; echo-back and line feed on crt
C11B 3E0A               LD    A,0AH             ;
C11D CD5702             CALL  0257H             ;
C120 F1                 POP   AF                ;
                ;
C121 FE63               CP    'c'
C123 2829               JR    Z,DSPLAY
C125 FE43               CP    'C'
C127 2825               JR    Z,DSPLAY
C129 FE64               CP    'd'
C12B 2821               JR    Z,DSPLAY
C12D FE44               CP    'D'
C12F 281D               JR    Z,DSPLAY
                ;
C131 FE6C               CP    'l'
C133 280E               JR    Z,LPRINT
C135 FE4C               CP    'L'
C137 280A               JR    Z,LPRINT
C139 FE70               CP    'p'
C13B 2806               JR    Z,LPRINT
C13D FE50               CP    'P'
```

```
C13F 2802               JR   Z,LPRINT
               ;
C141 18BD               JR   START
               ;
C143 2179C1    LPRINT:LD    HL,CARE
C146 CD59C1             CALL DSPMSG
               ;
C149 3E01               LD   A,1
C14B 3249EB             LD   (0EB49H),A        ; select printer for output
               ;
               ; output characters section
               ;
C14E 3E20      DSPLAY:LD    A,20H
               ;
C150 DF        LOOP:    RST  18H               ;*output to each device
C151 3C                 INC  A
C152 FE00               CP   0FFH+1
C154 20FA               JR   NZ,LOOP
               ;
C156 C3665C             JP   5C66H             ; jump to system monitor
               ;
               ; display message subroutine
               ;
C159 7E        DSPMSG:LD    A,(HL)
C15A A7                 AND  A
C15B C8                 RET  Z
C15C CD5702             CALL 0257H             ; display a character
C15F 23                 INC  HL
C160 18F7               JR   DSPMSG
               ;
               ; message area
               ;
C162 63727420  PROMPT:DEFB ´crt or line-printer ? ´,0
C166 6F72206C
C16A 696E652D
C16E 7072696E
C172 74657220
C176 3F2000
C179 6C696E65  CARE:    DEFB ´line-printer active !!´,0AH,0DH,0
C17D 2D707269
C181 6E746572
C185 20616374
C189 69766520
C18D 21210A0D
C191 00
               ;
C192                    END
```

```
*GC100
crt or line-printer ? l
line-printer active !!

*GC100
crt or line-printer ? c
 !"#$%&´()*+,-./0123456789:;<=>?@ABCDEFG
HIJKLMNOPQRSTUVWXYZ[¥]^_`abcdefghijklmno
pqrstuvwxyz{¦}~ ▁▂▃▄▅▆▇█▏▎▍▌▋▊▉┼┴┬┤├▔─▕│
┌ ┘ ╭ ╯ 。「」、・ヲァィゥェォャュョッーアイウエオカキクケコサシスセソ
タチツテトナニヌネノハヒフヘホマミムメモヤユヨラリルレロワン゛゜=╞╪▲▼
♠♥♦♣●○╱╲╳円年月日時分秒
*
```

００２ＢＨ　プリンタへの１バイト出力

アドレス　００２ＢＨ

機　　能　プリンタへの１バイト出力

レジスタ　－－－－－－－－－

解　　説　０Ｄ６０Ｈ番地へのジャンプを行うため，アキュムレータに格納されたデータ１バイトをプリンタへ出力します．

サンプル

```
                 ;
                 ; --- output message to printer ---
                 ;
                         ORG  0C100H
                 ;
C100 210EC1              LD   HL,MESAGE
                 ;
C103 7E          LOOP:   LD   A,(HL)            ; get a character
C104 A7                  AND  A                 ; check end mark or not end mark
C105 CA665C              JP   Z,5C66H           ; system monitor if end mark
                 ;
C108 CD2B00              CALL 002BH             ;*output a character to printer
C10B 23                  INC  HL                ; increment pointer
C10C 18F5                JR   LOOP
                 ;
                 ; message area
                 ;
C10E 2D2D2054    MESAGE:DEFB '-- This is the message !! --',0DH,0
C112 68697320
C116 69732074
C11A 6865206D
C11E 65737361
C122 67652021
C126 21202D2D
C12A 0D00
                 ;
C12C                     END
```

```
*GC100
-- This is the message !! --
*
```

ＰＣ－８００１ｍｋⅡのＮ－ＢＡＳＩＣ

ＰＣ－８００１のために開発されたＮ－ＢＡＳＩＣは，その機能，汎用性，操作性等を高く評価されることにより，１９８３年１０月現在ではＰＣ－８８０１そしてＰＣ－８００１ｍｋⅡのためにインプリメントされ，いずれも標準的に実装されています．

Ｎ－ＢＡＳＩＣがＰＣ－８００１に搭載されたものであるが，ＰＣ－８００１ｍｋⅡに搭載されたものであるのかは，１７５ＦＨ番地の内容によって判断することができます．すなわち，１７５ＦＨ番地の内容を調べ，それかＡＦＨであればＰＣ－８００１ｍｋⅡに搭載されたＮ－ＢＡＳＩＣ，ＡＦＨ以外（通常は８０Ｈ）であればＰＣ－８００１またはＰＣ－８８０１に搭載されたＮ－ＢＡＳＩＣであることがわかります．

また，Ｎ－ＢＡＳＩＣがＰＣ－８００１ｍｋⅡ用インプリメントを受けるにあたって変更された主な箇所はつぎに述べる３点です．

ＰＣ－８００１ｍｋⅡにはバックアップ電池を含むカレンダ時計（μＰＤ１９９０Ｃ）を内蔵しています．カレンダ時計のバックアップ電池は，本体の電源を連続して４０時間以上投入しておくと完全に充電されます．充電後さらに電源を投入し続けても，バックアップ電池の機能を損なうことはありません．完全に充電されたバックアップ電池は電源の切断後も，約２ヵ月間はカレンダ時計を動かし続けます．バックアップ電池を完全に充電しなくても，１日に３０分程度電源を投入しておけば，カレンダ時計は動き続けます．

ＰＣ－８００１にはこのようなカレンダ時計のバックアップ機能が用意されていなかったため，従来のＮ－ＢＡＳＩＣはホット・スタート時にカレンダ時計のイニシアライズ（７９年０１月０１日００時００分００秒に設定する）を行っていました．ＰＣ－８００１ｍｋⅡおよびＰＣ－８８０１のＮ－ＢＡＳＩＣでは，カレンダ時計のイニシアライズを行わないように変更されています．

さらにＰＣ－８００１ｍｋⅡのＮ－ＢＡＳＩＣでは，内蔵のＲＳ－２３２Ｃチャンネルに対しても，ＩＮＩＴ％，ＩＮＰＵＴ％，ＰＲＩＮＴ％がサポートされています．これらの命令を本体内蔵のＲＳ－２３２Ｃチャンネルに対して実行する場合には，ポート番号として１を指定します．

キーボード入力に関するルーチンも若干の変更がなされ，ＰＣ－８００１ｍｋⅡに新しく付加されたＴＡＢキーが有効になりました．

００３５Ｈ ＣＲＴへの１バイト出力

アドレス ００３５Ｈ

機　能 ＣＲＴへの１バイト出力

レジスタ ーーーーーーーーー

解　説 ０２５７Ｈ番地へのジャンプを行うため，アキュムレータに格納されたデータ１バイトを，ＣＲＴスクリーン上の現カーソル位置に出力します．

データが２０Ｈ以上の場合には通常のキャラクタとしてスクリーン上に表示しますが，２０Ｈ未満のコントロール・コードであればＣＲＴスクリーンのクリアやカーソルの移動等，コントロール・コードとしての動作を行います．

サンプル

```
                  ;
                  ; --- utility, search memory program ---
                  ;
                         ORG  0C100H
                  ;
                  ; input key-words section
                  ;
C100 21FBC1       START: LD   HL,TABLE
C103 0E00                LD   C,0              ; clear key-word counter
                  ;
C105 11EDC1       NWORD: LD   DE,PROMPT
C108 CDE4C1              CALL DSPMSG           ; display the prompt message
                  ;
C10B CD72C1              CALL INCOD
C10E 05                  DEC  B
C10F 281F                JR   Z,EIN            ; end of all input key-words
                  ;
C111 1003                DJNZ TWOCLM
C113 5A                  LD   E,D
C114 1630                LD   D,'0'            ; clear msb byte of key-word
                  ;
C116 7A           TWOCLM:LD   A,D
C117 FE22                CP   '"'
C119 7B                  LD   A,E
C11A 280D                JR   Z,EWORD
                  ;
C11C CDB8C1              CALL HEXBIN           ; convert msb byte into binary
C11F 47                  LD   B,A
                  ;
C120 7A                  LD   A,D
C121 CDB8C1              CALL HEXBIN           ; convert lsb byte into binary
                  ;
C124 07                  RLCA
C125 07                  RLCA
C126 07                  RLCA
C127 07                  RLCA
C128 80                  ADD  A,B
                  ;
C129 77           EWORD: LD   (HL),A           ; store a key-word
C12A 23                  INC  HL
C12B 0C                  INC  C                ; increment byte counter
C12C 38D2                JR   C,START          ; number of key-words overflow
C12E 18D5                JR   NWORD
                  ;
C130 3E0D         EIN:   LD   A,0DH
C132 CD3500              CALL 0035H            ;*display a carriage return code
```

```
C135 3E0A               LD   A,0AH
C137 CD3500             CALL 0035H               ;*display a line feed code
                ;
C13A 0C                 INC  C                   ;
C13B 0D                 DEC  C                   ; system monitor if no key-word
C13C CA665C             JP   Z,5C66H             ;
                ;
                ; search main  section
                ;
C13F 210000             LD   HL,0000H            ; set top address of search area
C142 11FF7F     IFEND:  LD   DE,07FFFH           ; set end address of search area
                ;
C145 E5                 PUSH HL
C146 A7                 AND  A                   ; reset carry flag
C147 ED52               SBC  HL,DE
C149 E1                 POP  HL
C14A 28B4               JR   Z,START             ; end of search
                ;
C14C C5                 PUSH BC
C14D E5                 PUSH HL
C14E 11FBC1             LD   DE,TABLE
                ;
C151 1A         MLOOP:  LD   A,(DE)
C152 BE                 CP   (HL)
C153 2007               JR   NZ,NXTMEM           ; next memory if not same
                ;
C155 0D                 DEC  C
C156 2804               JR   Z,NXTMEM            ; next memory if end of key-word
                ;
C158 13                 INC  DE
C159 23                 INC  HL
C15A 18F5               JR   MLOOP
                ;
C15C E1         NXTMEM:POP   HL
C15D C1                 POP  BC
C15E 23                 INC  HL
C15F 20E1               JR   NZ,IFEND
                ;
C161 2B         FOUND:  DEC  HL
C162 7C                 LD   A,H
C163 CDC4C1             CALL DSPACC              ; display the msb byte of address
C166 7D                 LD   A,L
C167 CDC4C1             CALL DSPACC              ; display the lsb byte of address
C16A 3E20               LD   A,' '
C16C CD3500             CALL 0035H               ;*display a space
C16F 23                 INC  HL
C170 18D0               JR   IFEND
                ;
                ; input a key-word subroutine
                ;
                ;    outputs  : input character in register de
                ;               input clumn plus one in register c
                ;    destroys : register a,f,b,d,e
                ;
C172 0601       INCOD:  LD   B,1
C174 CD750F             CALL 0F75H               ; input a character from keyboard
C177 FE0D               CP   0DH
C179 C8                 RET  Z                   ; end of input if return key
                ;
C17A FE7F               CP   7FH
C17C 28F4               JR   Z,INCOD
                ;
C17E FE20               CP   20H
C180 38F0               JR   C,INCOD             ; cancel control code
                ;
C182 CD3500             CALL 0035H               ;*echo-back the input character
C185 57                 LD   D,A                 ; store it in register d
                ;
C186 0602       INCOD1:LD    B,2
C188 CD750F             CALL 0F75H               ; input a character from keyboard
C18B FE0D               CP   0DH
C18D C8                 RET  Z                   ; end of input if return key
                ;
C18E FE08               CP   08H
```

```
C190 2804            JR   Z,DEL1
C192 FE7F            CP   7FH
C194 2005            JR   NZ,INCOD2
                ;
C196 CDDBC1     DEL1: CALL BACK          ; display a back-space
C199 18D7            JR   INCOD
                ;
C19B FE20       INCOD2:CP 20H
C19D 38E7            JR   C,INCOD1       ; cancel control code
                ;
C19F CD3500          CALL 0035H          ;*echo-back the input character
C1A2 5F              LD   E,A
                ;
C1A3 0603       INCOD3:LD B,3
C1A5 CD750F          CALL 0F75H          ; input a character from keyboard
C1A8 FE0D            CP   0DH
C1AA C8              RET  Z              ; end of input if return key
                ;
C1AB FE08            CP   08H
C1AD 2804            JR   Z,DEL2
C1AF FE7F            CP   7FH
C1B1 20F0            JR   NZ,INCOD3
                ;
C1B3 CDDBC1     DEL2: CALL BACK          ; display a back-space
C1B6 18CE            JR   INCOD1
                ;
                ; convert hexa code into binary code subroutine
                ;
                ;    inputs   : character code in accumulator
                ;    outputs  : binary code in accumulator
                ;    destroys : register a,f
                ;
C1B8 FE3A       HEXBIN:CP '9'+1
C1BA 3805            JR   C,HEXBI1
C1BC E60F            AND  0FH
C1BE C609            ADD  A,9
C1C0 C9              RET
                ;
C1C1 E60F       HEXBI1:AND 0FH
C1C3 C9              RET
                ;
                ; display accumulator hexa decimal subroutine
                ;
                ;    destroys : register a,f,b
                ;
C1C4 47         DSPACC:LD B,A
C1C5 0F              RRCA
C1C6 0F              RRCA
C1C7 0F              RRCA
C1C8 0F              RRCA
C1C9 CDCDC1          CALL HALF
C1CC 78              LD   A,B
                ;
C1CD E60F       HALF: AND 0FH
C1CF FE0A            CP   10
C1D1 3802            JR   C,NUMBER
C1D3 C607            ADD  A,7
                ;
C1D5 C630       NUMBER:ADD A,'0'
C1D7 CD3500          CALL 0035H          ;*display a character
C1DA C9              RET
                ;
                ; display a back-space subroutine
                ;
                ;    destroys : register a,f
                ;
C1DB D5         BACK: PUSH DE
C1DC 11F7C1          LD   DE,BAKSPC
C1DF CDE4C1          CALL DSPMSG
C1E2 D1              POP  DE
C1E3 C9              RET
                ;
                ; display message subroutine
                ;
```

```
                ;     inputs   : top address of message in register de
                ;     destroys : register a,f,d,e
                ;
C1E4 1A         DSPMSG:LD   A,(DE)
C1E5 A7                AND  A
C1E6 C8                RET  Z
C1E7 CD3500            CALL 0035H           ;*display a character
C1EA 13                INC  DE
C1EB 18F7              JR   DSPMSG
                ;
                ; message area
                ;
C1ED 0D0A636F PROMPT:DEFB 0DH,0AH,'code - ',0
C1F1 6465202D
C1F5 2000
C1F7 1D201D00 BAKSPC:DEFB 1DH,' ',1DH,0
                ;
                ; key-word table
                ;
C1FB            TABLE: END


*GC100

code - 'N
code - 'E
code - 'C
code -
1838 6108 6152
code - 'O
code - 'k
code - fc
code - 0d
code - 0a
code - 00
code -
3B60
code - 21
code - 00
code - 00
code -
047A 0C6D 1540 2306 27E9 28B6 290A 295F
5055 516C 5335 5D1B 5D68 5D74 6405 66CF
6BFD 762B 7859 78D0 796F 7A87
code - 0
code - 0
code - 0
code - 0
code - 0
code - 0
code - 0
code - 0
code - 0
code - 0
code -
0046 0047 0048 0049 004A 004B 004C 004D
004E 004F 0050 0051 0052 0053 0054 0055
0056 0057 0058 0059 005A 005B 005C 005D
005E 005F 0060 3A71 3A72 3A73 3A74 3A82
3A83 3A84 3A85 3A86 3A87 3A96 3A97 3AA7
3AB6 3AB7 3AC5 3AC6 3AC7 3AD6 3AD7 3AE5
3AE6 3AE7 3AF7 3B16 3B17 3B18 444B 7FDB
7FDC 7FDD 7FDE 7FDF 7FE0 7FE1 7FE2 7FE3
7FE4 7FE5 7FE6 7FE7 7FE8 7FE9 7FEA 7FEB
7FEC 7FED 7FEE 7FEF 7FF0 7FF1 7FF2 7FF3
7FF4 7FF5 7FF6
code -

*
```

００６ＡＨ　ホット・スタート

アドレス	００６ＡＨ
機　能	ホット・スタート
レジスタ	ＡＦＢＣＤＥＨＬ

解　説　ＣＲＴスクリーン，タイマ等，インタフェース関係のイニシアライズを実行後，インタプリンタの制御下に入ります．

ただし，ＢＡＳＩＣのテキストは保証します．

サンプル

```
                ;
                ; --- cold start or hot start ---
                ;
                        ORG   0C100H
                ;
C100 215EC1     START:  LD    HL,PROMPT
C103 CD55C1             CALL  DSPMSG
                ;
C106 CD750F     LOOP:   CALL  0F75H             ; input a character from keyboard
C109 CBAF               RES   5,A               ; convert into capital letter
C10B FE43               CP    'C'
C10D 280A               JR    Z,COLD
C10F FE48               CP    'H'
C111 2824               JR    Z,HOT
C113 FE4E               CP    'N'
C115 2829               JR    Z,NOT
C117 18ED               JR    LOOP
                ;
C119 217BC1     COLD:   LD    HL,MSGC           ;
C11C CD55C1             CALL  DSPMSG            ;
C11F 060A               LD    B,10              ;
C121 3E3E       COLD1:  LD    A,'>'             ;
C123 CD5702             CALL  0257H             ;
C126 CD49C1             CALL  WAIT              ; demonstration before cold start
C129 10F6               DJNZ  COLD1             ;
C12B 2191C1             LD    HL,MSGJ           ;
C12E CD55C1             CALL  DSPMSG            ;
C131 CD49C1             CALL  WAIT              ;
                ;
C134 C30000             JP    0000H             ; just cold start
                ;
C137 219AC1     HOT:    LD    HL,MSGH
C13A CD55C1             CALL  DSPMSG
C13D C36A00             JP    006AH             ;*hot start


C140 21B1C1     NOT:    LD    HL,MSGN
C143 CD55C1             CALL  DSPMSG
C146 C3665C             JP    5C66H             ; system monitor
                ;
                ; wait subroutine
                ;
                ;    destroys : flags only
                ;
C149 C5         WAIT:   PUSH  BC
C14A 0EFF               LD    C,0FFH
C14C 06FF       WAIT1:  LD    B,0FFH
C14E 10FE       WAIT2:  DJNZ  WAIT2
C150 0D                 DEC   C
C151 20F9               JR    NZ,WAIT1
C153 C1                 POP   BC
```

```
C154 C9               RET
                ;
                ; display message subroutine
                ;
                ;    inputs   : top address of message in register hl
                ;    destroys : register a,f,h,l
                ;
C155 7E         DSPMSG:LD   A,(HL)
C156 A7               AND  A
C157 C8               RET  Z
C158 CD5702           CALL 0257H            ; display a character
C15B 23               INC  HL
C15C 18F7             JR   DSPMSG
                ;
                ; message area
                ;
C15E 636F6C64 PROMPT:DEFB 'cold or hot start (c/h/n) ? ',0
C162 206F7220
C166 686F7420
C16A 73746172
C16E 74202863
C172 2F682F6E
C176 29203F20
C17A 00
C17B 633A6578 MSGC:  DEFB 'c:execute cold start ',0
C17F 65637574
C183 6520636F
C187 6C642073
C18B 74617274
C18F 2000
C191 206A7573 MSGJ:  DEFB ' just !!',0
C195 74202121
C199 00
C19A 683A6578 MSGH:  DEFB 'h:execute hot start !!',0
C19E 65637574
C1A2 6520686F
C1A6 74207374
C1AA 61727420
C1AE 212100
C1B1 6E3A646F MSGN:  DEFB 'n:do nothing !!',0
C1B5 206E6F74
C1B9 68696E67
C1BD 20212100
                ;
C1C1                  END
```

```
*GC100
cold or hot start (c/h/n) ? n:do nothing !!
*GC100
cold or hot start (c/h/n) ? h:execute hot start !!
Ok
mon
*GC100
cold or hot start (c/h/n) ? c:execute cold start >>>>>>>>>> just !!
```

００８１H　BASICのコマンド待ち

アドレス	００８１H
機　能	BASICのコマンド待ち
レジスタ	AFBCDEHL

解　説　インタフェース関係のイニシアライズ等を全く行わずに，現在のカーソル・ポジションからプロンプト・メッセージ“OK”を表示後，ＢＡＳＩＣのスクリーン・エディット・モードに入ります．

ただし，ＢＡＳＩＣのテキストは保証します．

００８１Ｈ番地からＦ１Ｂ０Ｈ番地へのフックが用意されています．

サンプル

```
               ;
               ; --- select each start ---
               ;
                      ORG   0C100H
               ;
C100 216BC1    START: LD    HL,PROMPT
C103 CD62C1           CALL  DSPMSG
               ;
C106 CD750F    LOOP:  CALL  0F75H              ; input a character from keyboard
C109 CBAF             RES   5,A                ; convert into capital letter
C10B FE43             CP    'C'
C10D 280E             JR    Z,COLD
C10F FE48             CP    'H'
C111 2828             JR    Z,HOT
C113 FE42             CP    'B'
C115 282D             JR    Z,BAS
C117 FE4E             CP    'N'
C119 2832             JR    Z,NOT
C11B 18E9             JR    LOOP
               ;
C11D 2187C1    COLD:  LD    HL,MSGC            ;
C120 CD62C1           CALL  DSPMSG             ;
C123 060A             LD    B,10               ;
C125 3E3E      COLD1: LD    A,'>'              ;
C127 CD5702           CALL  0257H              ;
C12A CD56C1           CALL  WAIT               ; demonstration before cold start
C12D 10F6             DJNZ  COLD1              ;
C12F 219DC1           LD    HL,MSGJ            ;
C132 CD62C1           CALL  DSPMSG             ;
C135 CD56C1           CALL  WAIT               ;
               ;
C138 C30000           JP    0000H              ; just cold start
               ;
C13B 21A6C1    HOT:   LD    HL,MSGH
C13E CD62C1           CALL  DSPMSG
C141 C36A00           JP    006AH              ; hot start
               ;
C144 21BDC1    BAS:   LD    HL,MSGB
C147 CD62C1           CALL  DSPMSG
C14A C38100           JP    0081H              ;*under the basic
               ;
C14D 21D0C1    NOT:   LD    HL,MSGN
C150 CD62C1           CALL  DSPMSG
C153 C3665C           JP    5C66H              ; system monitor
               ;
               ; wait subroutine
```

```
                ;
                ;    destroys : flags only
                ;
C156 C5         WAIT:  PUSH BC
C157 0EFF              LD   C,0FFH
C159 06FF       WAIT1: LD   B,0FFH
C15B 10FE       WAIT2: DJNZ WAIT2
C15D 0D                DEC  C
C15E 20F9              JR   NZ,WAIT1
C160 C1                POP  BC
C161 C9                RET
                ;
                ; display message subroutine
                ;
                ;    inputs   : top address of message in register hl
                ;    destroys : register a,f,h,l
                ;
C162 7E         DSPMSG:LD   A,(HL)
C163 A7                AND  A
C164 C8                RET  Z
C165 CD5702            CALL 0257H            ; display a character
C168 23                INC  HL
C169 18F7              JR   DSPMSG
                ;
                ; message area
                ;
C16B 65616368 PROMPT:DEFB 'each selection (c/h/b/n) ? ',0
C16F 2073656C
C173 65637469
C177 6F6E2028
C17B 632F682F
C17F 622F6E29
C183 203F2000
C187 633A6578 MSGC:  DEFB 'c:execute cold start ',0
C18B 65637574
C18F 6520636F
C193 6C642073
C197 74617274
C19B 2000
C19D 206A7573 MSGJ:  DEFB ' just !!',0
C1A1 74202121
C1A5 00
C1A6 683A6578 MSGH:  DEFB 'h:execute hot start !!',0
C1AA 65637574
C1AE 6520686F
C1B2 74207374
C1B6 61727420
C1BA 212100
C1BD 623A6A75 MSGB:  DEFB 'b:jump to basic !!',0
C1C1 6D702074
C1C5 6F206261
C1C9 73696320
C1CD 212100
C1D0 6E3A646F MSGN:  DEFB 'n:do nothing !!',0
C1D4 206E6F74
C1D8 68696E67
C1DC 20212100
                ;
C1E0                   END
```

```
*GC100
each selection (c/h/b/n) ? n:do nothing !!
*GC100
each selection (c/h/b/n) ? b:jump to basic !!
Ok
mon
*GC100
each selection (c/h/b/n) ? h:execute hot start !!
Ok
mon
*GC100
each selection (c/h/b/n) ? c:execute cold start >>>>>>>>>> just !!
```

００９９Ｈ　ＰＣ－８０３１との１セクタ入出力

アドレス	００９９Ｈ
機　能	ＰＣ－８０３１との１セクタ入出力
レジスタ	－Ｆ－－－－－－

解　説　ミニ・フロッピ・ディスク・ユニット（ＰＣ－８０３１，ＰＣ－８０３１－２Ｗ，ＰＣ－８０Ｓ３１相当）との入出力を行うための最も基本的でかつ有効なシステム・サブルーチンで，ＰＣ－８００１ｍｋⅡのメモリ上からミニ・フロッピへの書き込み，逆にミニ・フロッピからメモリ上への読み込み，そしてデータのチェックを行うことができます．

システム・サブルーチンの呼び出しに際しては，次に示す入力パラメータを正確に与えることが必要です．

ＥＤＣ５Ｈ番地←０
ＥＤＣ６Ｈ番地←０から始まるドライブ番号
レジスタＢ　　←０から始まるトラック番号
レジスタＣ　　←１から始まるセクタ番号
アキュムレータ←入出力実行セクタ数
レジスタＤＥ　←データ格納メモリの開始アドレス

書き込み，読み出し，チェックの指定はフラグによって行います．

書き込みの場合，キャリ・フラグ（ＣＹ）をセットします．
読み出しの場合，ゼロ・フラグ（Ｚ）をリセットします．
チェックの場合，ゼロ・フラグ（Ｚ）をセットします．

エラーが発生した場合にはキャリ・フラグ（ＣＹ）をセットして，そうでなければキャリ・フラグをリセットしてもどります．さらに，エラーの発生ごとにＥＤＣ５Ｈ番地の値をインクリメントし，エラーが１０回を越えた時点で，“Disk I／O error”エラーが発生し，インタプリタの制御下に入ります．

また，指定したドライブが両面仕様の場合，トラック番号が偶数の時には表面をアクセスし，トラック番号が奇数の時には裏面をアクセスするため，サーフェイスの指定は必要ありません．

サンプル

```
                ;
                ; --- control mini floppy disk ---
                ;
                        ORG   0C100H
                ;
C100 2191C1             LD    HL,MSGDR
C103 CD88C1             CALL  DSPMSG
C106 CD6CC1             CALL  INDEC
C109 3D                 DEC   A
C10A 32C6ED             LD    (0EDC6H),A        ; set drive number
                ;
C10D 219DC1             LD    HL,MSGTR
C110 CD88C1             CALL  DSPMSG
C113 CD6CC1             CALL  INDEC
C116 47                 LD    B,A               ; set track number
                ;
C117 21A9C1             LD    HL,MSGSEC
C11A CD88C1             CALL  DSPMSG
C11D CD6CC1             CALL  INDEC
C120 4F                 LD    C,A               ; set sector number
                ;
C121 21B5C1             LD    HL,MSGWRV
C124 CD88C1             CALL  DSPMSG
C127 CD750F             CALL  0F75H             ; input a character from keyboard
C12A CD5702             CALL  0257H             ; echo-back a character
                ;
C12D FE77               CP    'w'
C12F 2828               JR    Z,WRITE
C131 FE57               CP    'W'
C133 2824               JR    Z,WRITE
C135 FE72               CP    'r'
C137 2823               JR    Z,READ
C139 FE52               CP    'R'
C13B 281F               JR    Z,READ
C13D FE76               CP    'v'
C13F 281F               JR    Z,VERIFY
C141 FE56               CP    'V'
C143 281B               JR    Z,VERIFY
C145 181C               JR    ERROR
                ;
C147 3E00       CTRLDS:LD     A,0
C149 32C5ED             LD    (0EDC5H),A
C14C 3E01               LD    A,1
C14E 1100C2             LD    DE,BUFFER         ; set top address of buffer
C151 CD9900             CALL  0099H             ;*just control disk
C154 380D               JR    C,ERROR
C156 C3665C             JP    5C66H             ; jump to system monitor
                ;
C159 37         WRITE:  SCF                   ; set carry flag
C15A 18EB               JR    CTRLDS
                ;
C15C AF         READ:   XOR   A                 ; reset carry flag
C15D 3C                 INC   A                 ; reset zero flag
C15E 18E7               JR    CTRLDS
                ;
C160 AF         VERIFY:XOR    A                 ; reset carry and set zero flag
C161 18E4               JR    CTRLDS
                ;
C163 21D0C1     ERROR:  LD    HL,MSGERR         ;
C166 CD88C1             CALL  DSPMSG            ; exit with error
C169 C3665C             JP    5C66H             ;
                ;
                ; input decimal number subroutine
                ;
                ;    outputs  : input number in accumulator
                ;    destroys : register a,f,e
                ;
C16C 1E00       INDEC:  LD    E,0               ; initialize input number
C16E 3E30               LD    A,'0'             ; initialize input character
                ;
C170 F5         INLOOP:PUSH   AF                ;
```

```
C171 7B                 LD   A,E              ;
C172 87                 ADD  A,A              ;
C173 87                 ADD  A,A              ;
C174 83                 ADD  A,E              ;
C175 87                 ADD  A,A              ; e <-- e * 10 + input number
C176 5F                 LD   E,A              ;
C177 F1                 POP  AF               ;
C178 D630               SUB  '0'              ;
C17A 83                 ADD  A,E              ;
C17B 5F                 LD   E,A              ;
                ;
C17C CD750F             CALL 0F75H            ; input a character from keyboard
C17F CD5702             CALL 0257H            ; echo-back a character
                ;
C182 FE0D               CP   0DH
C184 20EA               JR   NZ,INLOOP
                ;
C186 7B                 LD   A,E
C187 C9                 RET
                ;
                ; display message subroutine
                ;
C188 7E         DSPMSG:LD    A,(HL)
C189 A7                 AND  A
C18A C8                 RET  Z
C18B CD5702             CALL 0257H            ; display a character
C18E 23                 INC  HL
C18F 18F7               JR   DSPMSG
                ;
                ; message area
                ;
C191 0D0A6472 MSGDR: DEFB 0DH,0AH,'drive  ? ',0
C195 69766520
C199 203F2000
C19D 0D0A7472 MSGTR: DEFB 0DH,0AH,'track  ? ',0
C1A1 61636B20
C1A5 203F2000
C1A9 0D0A7365 MSGSEC:DEFB 0DH,0AH,'sector ? ',0
C1AD 63746F72
C1B1 203F2000
C1B5 0D0A7772 MSGWRV:DEFB 0DH,0AH,'write, read or verify ? ',0
C1B9 6974652C
C1BD 20726561
C1C1 64206F72
C1C5 20766572
C1C9 69667920
C1CD 3F2000
C1D0 0D0A6572 MSGERR:DEFB 0DH,0AH,'error !!',07H,0
C1D4 726F7220
C1D8 21210700
                ;
                ; equate buffer area
                ;
C200            BUFFER:EQU  0C200H
                ;
C1DC                    END
```

```
*GC100

drive  ? 1
track  ? 0
sector ? 1
write, read or verify ? r
*DC200,C2FF
C200 3E 0B CD 7C 01 3E 07 CD 83 01 3E EF CD 83 01 AF
C210 CD 83 01 3E 01 CD 83 01 CD E9 01 2F E6 F0 FE 10
C220 20 0D 3E 17 CD 7C 01 3E 0F CD 83 01 C3 C2 C0 21
C230 38 C0 CD ED 52 C3 BC C0 4E 4F 4E 20 44 49 53 4B
C240 20 42 41 53 49 43 0D 0A 00 F5 79 2F 3C C6 11 F5
C250 3E EA 92 FE 10 30 02 E1 F5 F1 B7 CD 99 00 38 05
C260 82 57 F1 B7 C9 F1 37 C9 F1 F1 F5 32 B9 ED 3E 02
C270 CD 21 01 F1 F5 47 CA C2 00 CD 5A 01 E6 40 CA 68
C280 01 CD 87 C0 C3 C2 00 3E 12 CD 7C 01 0E 80 CD 97
C290 C0 0D 20 FA 10 F6 C9 3E 0B D3 FF DB FE E6 01 28
C2A0 FA 3E 0A D3 FF DB FC 12 13 3E 0D D3 FF DB FE E6
C2B0 01 20 FA DB FC 12 13 3E 0C D3 FF C9 AF 32 C7 ED
C2C0 37 C9 11 00 C1 01 02 00 AF 32 C5 ED 3E 01 B7 CD
C2D0 99 00 3E C3 32 B3 F1 21 68 C0 22 B4 F1 11 00 C2
C2E0 01 03 00 AF 32 C5 ED CD 49 C0 38 D0 04 04 0E 01
C2F0 CD 49 C0 38 C7 04 04 0E 01 CD 49 C0 38 BE 11 A5
*GC100

drive  ? 2
track  ? 0
sector ? 1
write, read or verify ? r
*DC200,C2FF
C200 21 00 84 AF 32 B4 EC 01 02 00 11 11 2D 3A 5D EF
C210 B7 20 04 06 02 1E 1B AF 3C E5 D5 C5 CD 9A 36 C1
C220 D1 E1 30 08 3A B4 EC FE 03 20 EC C9 AF 32 B4 EC
C230 15 28 0B 0C 24 7B B9 20 DE 04 0E 01 18 D9 CD 00
C240 84 C3 00 88 22 E8 E7 C9 06 28 21 60 C0 5E 23 56
C250 23 4E 23 7E 23 EB 36 C3 23 71 23 77 EB 10 EE C9
C260 DC EC AC D9 C4 EC A2 D0 C1 EC DE D9 D0 EC 14 D4
C270 AE ED 8E D9 E8 EC A4 D4 B1 ED 4A CE EE EC EB D4
C280 F1 EC 6F D9 F4 EC 82 D9 6F ED FE CF AB ED E1 CE
C290 FF ED 76 CE BB EC 4B C3 D3 EC D8 ED D6 EC D8 ED
C2A0 9F ED D8 ED A2 ED D8 ED A2 ED A5 ED A8 ED D8 ED
C2B0 74 EE D8 ED 77 EE D8 ED 7A EE E0 DB 7D EE 22 C2
C2C0 80 EE CC DA 83 EE F6 DA 86 EE A5 DE 89 EE 6D DB
C2D0 8C EE 56 D2 8F EE 96 CE 92 EE 8A D0 95 EE 08 D5
C2E0 98 EE 36 D5 9B EE 49 D3 9E EE 44 D3 BC EE 00 C1
C2F0 BF EE 23 DA C2 EE E7 D9 C5 EE A2 DE C8 EE EC C2
*GC100

drive  ? 1
track  ? 0
sector ? 1
write, read or verify ? x
error !!
*
```

００ＣＢＨ　拡張システム・チェック

アドレス	００ＣＢＨ
機　能	拡張システム・チェック
レジスタ	ＡＦーーーーーー

解　説　ディスク・ユニット等の拡張システムが接続されていればゼロ・フラグ（Ｚ）をリセットして，接続されていなければゼロ・フラグ（Ｚ）をセットしてもどります.

拡張システム接続の有無は，入力ポート４０Ｈ番地からの入力データによって判断します.

サンプル

```
                ;
                ; --- check extended system or not ---
                ;
                        ORG   0C100H
                ;
C100 CDCB00             CALL  00CBH              ;*check extended system or not
C103 2809               JR    Z,NOTEXT
                ;
C105 2120C1     YESEXT:LD     HL,MSGYES          ;
C108 CD17C1             CALL  DSPMSG             ; yes extended system
C10B C3665C             JP    5C66H              ;
                ;
C10E 2137C1     NOTEXT:LD     HL,MSGNOT          ;
C111 CD17C1             CALL  DSPMSG             ; not extended system
C114 C3665C             JP    5C66H              ;
                ;
                ; display message subroutine
                ;
C117 7E         DSPMSG:LD     A,(HL)
C118 A7                 AND   A
C119 C8                 RET   Z
C11A CD5702             CALL  0257H              ; display a character
C11D 23                 INC   HL
C11E 18F7               JR    DSPMSG
                ;
                ; message area
                ;
C120 79657320   MSGYES:DEFB 'yes extended system !!',0
C124 65787465
C128 6E646564
C12C 20737973
C130 74656D20
C134 212100
C137 6E6F7420   MSGNOT:DEFB 'not extended system !!',0
C13B 65787465
C13F 6E646564
C143 20737973
C147 74656D20
C14B 212100
                ;
C14E                    END


*GC100
yes extended system !!
*
```

０２５７Ｈ　ＣＲＴへの１バイト出力

アドレス	０２５７Ｈ
機　　能	ＣＲＴへの１バイト出力
レジスタ	ーーーーーーーーー

解　　説　アキュムレータに格納されたデータ１バイトを，ＣＲＴスクリーン上の現カーソル位置に出力します.

データが２０Ｈ以上の場合には通常のキャラクタとしてスクリーン上に表示しますが，２０Ｈ未満のコントロール・コードであればＣＲＴスクリーンのクリアやカーソルの移動等，コントロール・コードとしての動作を行います.

サンプル

```
                ;
                ; --- display characters ---
                ;
                        ORG   0C100H
                ;
C100 3E20               LD    A,20H
                ;
C102 CD5702     LOOP:   CALL  0257H             ;*output a character to crt screen
C105 3C                 INC   A
C106 FE00               CP    0FFH+1
C108 20F8               JR    NZ,LOOP
                ;
C10A C3665C             JP    5C66H             ; jump to system monitor
                ;
C10D                    END


*GC100
 !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFG
HIJKLMNOPQRSTUVWXYZ[¥]^_`abcdefghijklmno
pqrstuvwxyz{¦}~ ▁▂▃▄▅▆▇█▏▎▍▌▋▊▉┼┴┬┤├▔▁▕▏
┌┐└┘╭╮╰╯ ｡｢｣､･ｦｧｨｩｪｫｬｭｮｯｰｱｲｳｴｵｶｷｸｹｺｻｼｽｾｿ
ﾀﾁﾂﾃﾄﾅﾆﾇﾈﾉﾊﾋﾌﾍﾎﾏﾐﾑﾒﾓﾔﾕﾖﾗﾘﾙﾚﾛﾜﾝﾞﾟ═╪╪▲▼
♠♥♦♣●○╱╲╳円年月日時分秒
*
```

０２Ｄ７Ｈ　カーソル移動とＣＲＴへの１バイト出力

アドレス　０２Ｄ７Ｈ
機　能　カーソル移動とＣＲＴへの１バイト出力
レジスタ　ＡＦＢＣＤＥＨＬ

解　説　レジスタＨに格納された１～８０の横座標，およびレジスタＬに格納された１～２５の縦座標，によって指定するＣＲＴスクリーン上の指定位置にカーソルを移動後，アキュムレータに格納されたデータ１バイトを，ＣＲＴスクリーン上のカーソル位置に出力します．

データが２０Ｈ以上の場合には通常のキャラクタとしてスクリーン上に表示しますが，２０Ｈ未満のコントロール・コードであればＣＲＴスクリーンのクリアやカーソルの移動等，コントロール・コードとしての動作を行います．

サンプル

```
               ;
               ; --- display steps on text screen ---
               ;
                       ORG  0C100H
               ;
C100 2E04              LD   L,4              ; appoint coordinate y
               ;
C102 3E50      LOOPY:  LD   A,80
C104 95                SUB  L
C105 67                LD   H,A              ; appoint coordinate x
C106 0630              LD   B,30H
               ;
C108 C5        LOOPX:  PUSH BC
C109 E5                PUSH HL
C10A 3EDB              LD   A,'□'            ; appoint a character
C10C CDD702            CALL 02D7H            ;*move cursor and display
C10F CD20C1            CALL WAIT             ; wait a moment
C112 E1                POP  HL
C113 C1                POP  BC
C114 25                DEC  H                ; decrement coordinate x
C115 10F1              DJNZ LOOPX
               ;
C117 2C                INC  L                ; increment coordinate y
C118 7D                LD   A,L
C119 FE1A              CP   25+1
C11B 20E5              JR   NZ,LOOPY
               ;
C11D C3665C            JP   5C66H            ; jump to system monitor
               ;
               ; wait subroutine
               ;
               ;    destroys : register c only
               ;
C120 0EFF      WAIT:   LD   C,0FFH
C122 0D        WAIT1:  DEC  C
C123 20FD              JR   NZ,WAIT1
C125 C9                RET
               ;
C126                   END
```

```
*GC100

*
                            □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
                           □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
                          □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
                         □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
                        □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
                       □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
                      □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
                     □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
                    □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
                   □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
                  □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
                 □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
                □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
               □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
              □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
             □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
            □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
           □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
          □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
         □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
        □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
       □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
```

０３５０Ｈ　ベル・コードの出力

アドレス　０３５０Ｈ

機　能　ベル・コードの出力

レジスタ　ＡＦ－－－ＥＨＬ

解　説　ＢＡＳＩＣのＢＥＥＰと同様に，内蔵ブザーを一定時間鳴動させます．すでに内蔵ブザーが鳴動している場合には，ブザーの鳴動を一定時間停止させます．

サンプル

```
                ;
                ; --- ring the bell each times ---
                ;
                        ORG   0C100H
                ;
C100 213CC1             LD    HL,MESAGE
C103 CD33C1             CALL  DSPMSG
                ;
C106 CD17C1             CALL  INDEC                ; input times
                ;
C109 3C                 INC   A                    ;
C10A 3D                 DEC   A                    ; jump to system monitor if zero
C10B CA665C             JP    Z,5C66H              ;
                ;
C10E 47                 LD    B,A
C10F CD5003     LOOP:   CALL  0350H                ;*ring the bell
C112 10FB               DJNZ  LOOP
                ;
C114 C3665C             JP    5C66H                ; jump to system monitor
                ;
                ; input decimal number subroutine
                ;
                ;    outputs  : input number in accumulator
                ;    destroys : register a,f,e
                ;
C117 1E00       INDEC:  LD    E,0                  ; initialize input number
C119 3E30               LD    A,'0'                ; initialize input character
                ;
C11B F5         INLOOP:PUSH   AF                   ;
C11C 7B                 LD    A,E                  ;
C11D 87                 ADD   A,A                  ;
C11E 87                 ADD   A,A                  ;
C11F 83                 ADD   A,E                  ;
C120 87                 ADD   A,A                  ; e <-- e * 10 + input number
C121 5F                 LD    E,A                  ;
C122 F1                 POP   AF                   ;
C123 D630               SUB   '0'                  ;
C125 83                 ADD   A,E                  ;
C126 5F                 LD    E,A                  ;
                ;
C127 CD750F             CALL  0F75H                ; input a character from keyboard
C12A CD5702             CALL  0257H                ; echo-back a character
                ;
C12D FE0D               CP    0DH
C12F 20EA               JR    NZ,INLOOP
                ;
C131 7B                 LD    A,E
C132 C9                 RET
                ;
                ; display message subroutine
                ;
```

```
C133 7E        DSPMSG:LD   A,(HL)
C134 A7               AND  A
C135 C8               RET  Z
C136 CD5702           CALL 0257H           ; display a character
C139 23               INC  HL
C13A 18F7             JR   DSPMSG
               ;
               ; message area
               ;
C13C 686F7720 MESAGE:DEFB 'how many times ? ',0
C140 6D616E79
C144 2074696D
C148 6573203F
C14C 2000
               ;
C14E                  END


*GC100
how many times ? 150
*
```

０３Ａ９Ｈ　カーソルの移動

アドレス	０３Ａ９Ｈ
機　　能	カーソルの移動
レジスタ	ＡＦ－－－－－－

解　　説　レジスタＨに格納された１～８０の横座標，およびレジスタＬに格納された１～２５の縦座標，によって指定するＣＲＴスクリーン上の指定位置にカーソルを移動します．

サンプル

```
                  ;
                  ; --- move cursor to appointment position ---
                  ;
                          ORG   0C100H
                  ;
C100 2601         LOOP:   LD    H,1            ;
C102 2E19                 LD    L,25           ;
C104 CDA903               CALL  03A9H          ; display prompt message-1
C107 2167C1               LD    HL,PRMPT1      ;
C10A CD5EC1               CALL  DSPMSG         ;
                  ;
C10D CD42C1               CALL  INDEC          ; input position x
                  ;
C110 3C                   INC   A
C111 67                   LD    H,A
C112 E5                   PUSH  HL
                  ;
C113 261D                 LD    H,29           ;
C115 2E19                 LD    L,25           ;
C117 CDA903               CALL  03A9H          ; display prompt message-2
C11A 2182C1               LD    HL,PRMPT2      ;
C11D CD5EC1               CALL  DSPMSG         ;
                  ;
C120 CD42C1               CALL  INDEC          ; input position y
                  ;
C123 3C                   INC   A
C124 E1                   POP   HL
C125 6F                   LD    L,A
C126 E5                   PUSH  HL
                  ;
C127 2639                 LD    H,57           ;
C129 2E19                 LD    L,25           ;
C12B CDA903               CALL  03A9H          ; display prompt message-3
C12E 219DC1               LD    HL,PRMPT3      ;
C131 CD5EC1               CALL  DSPMSG         ;
                  ;
C134 CD42C1               CALL  INDEC          ; input character code
                  ;
C137 E1                   POP   HL
                  ;
C138 F5                   PUSH  AF
C139 CDA903               CALL  03A9H          ; just move cursor
C13C F1                   POP   AF
                  ;
C13D CD5702               CALL  0257H          ; just display a character
                  ;
C140 18BE                 JR    LOOP           ; repeat
                  ;
                  ; input decimal number subroutine
                  ;
                  ;    outputs  : input number in accumulator
```

```
                ;     destroys : register a,f,e
                ;
C142 1E00       INDEC: LD   E,0              ; initialize input number
C144 3E30              LD   A,'0'            ; initialize input character
                ;
C146 F5         INLOOP:PUSH AF               ;
C147 7B                LD   A,E              ;
C148 87                ADD  A,A              ;
C149 87                ADD  A,A              ;
C14A 83                ADD  A,E              ;
C14B 87                ADD  A,A              ; e <-- e * 10 + input number
C14C 5F                LD   E,A              ;
C14D F1                POP  AF               ;
C14E D630              SUB  '0'              ;
C150 83                ADD  A,E              ;
C151 5F                LD   E,A              ;
                ;
C152 CD750F            CALL 0F75H            ; input a character from keyboard
C155 CD5702            CALL 0257H            ; echo-back a character
                ;
C158 FE0D              CP   0DH
C15A 20EA              JR   NZ,INLOOP
                ;
C15C 7B                LD   A,E
C15D C9                RET
                ;
                ; display message subroutine
                ;
C15E 7E         DSPMSG:LD   A,(HL)
C15F A7                AND  A
C160 C8                RET  Z
C161 CD5702            CALL 0257H            ; display a character
C164 23                INC  HL
C165 18F7              JR   DSPMSG
                ;
                ; message area
                ;
C167 6E657874 PRMPT1:DEFB 'next position x ?     ',1DH,1DH,1DH,1DH,0
C16B 20706F73
C16F 6974696F
C173 6E207820
C177 3F202020
C17B 20201D1D
C17F 1D1D00
C182 6E657874 PRMPT2:DEFB 'next position y ?     ',1DH,1DH,1DH,1DH,0
C186 20706F73
C18A 6974696F
C18E 6E207920
C192 3F202020
C196 20201D1D
C19A 1D1D00
C19D 63686172 PRMPT3:DEFB 'character code ?     ',1DH,1DH,1DH,1DH,0
C1A1 61637465
C1A5 7220636F
C1A9 6465203F
C1AD 20202020
C1B1 201D1D1D
C1B5 1D00
                ;
C1B7                   END
```

０３Ｄ９Ｈ　ＶＲＴＣのチェック

アドレス　０３Ｄ９Ｈ

機　　能　ＶＲＴＣのチェック

レジスタ　ＡＦ－－－－－－

解　　説　ＣＲＴがバーチカル・リトレース期間になるまでの間，同期待ちを行います．

ＶＲＴＣは，入力ポート４０Ｈ番地の第５ビットによって，チェックすることができます．

サンプル

```
                ;
                ; --- check vrtc ---
                ;
                        ORG  0C100H
                ;
C100 2120C1             LD   HL,MESAGE
C103 CD17C1             CALL DSPMSG
                ;
C106 CDD903     LOOP:   CALL 03D9H            ;*just check vrtc
                ;
C109 3E2A               LD   A,'*'
C10B CD5702             CALL 0257H
                ;
C10E DB09               IN   A,(09H)          ;
C110 CB47               BIT  0,A              ; jump to system monitor if stop key
C112 CA665C             JP   Z,5C66H          ;
                ;
C115 18EF               JR   LOOP
                ;
                ; display message subroutine
                ;
C117 7E         DSPMSG:LD    A,(HL)
C118 A7                 AND  A
C119 C8                 RET  Z
C11A CD5702             CALL 0257H            ; display a character
C11D 23                 INC  HL
C11E 18F7               JR   DSPMSG
                ;
                ; message area
                ;
C120 76727463 MESAGE:DEFB 'vrtc ready : ',0
C124 20726561
C128 6479203A
C12C 2000
                ;
C12E                    END
```

```
*GC100
vrtc ready : ***************************
****************************************
****************************************
****************************************
****************************************
****************************************
****************************************
****************************************
****************************************
*************
*
```

０３Ｆ３Ｈ　キャラクタ座標▷ＶＲＡＭアドレス変換

アドレス	０３Ｆ３Ｈ
機　能	キャラクタ座標▷ＶＲＡＭアドレス変換
レジスタ	ＡＦＢＣＤＥＨＬ

解　説　レジスタＨに格納された１～８０の横座標，およびレジスタＬに格納された１～２５の縦座標，によって指定するＣＲＴスクリーン上の指定位置に相当するＶＲＡＭ(Video－ＲＡＭ)のアドレスを計算し，結果をレジスタＨＬに格納してもどります．

８０／７２／４０／３６桁モード，カラー／白黒モード等はすべて考慮した上で，アドレスが計算されます．

なお，８０／７２／４０／３６桁モードの判別に当たってはワーク・エリア中ＥＡ６５Ｈ番地の値を，カラー／白黒モードの判別に当たってはＥＡ６１Ｈ番地の値を，それぞれ参照します．

サンプル

```
                ;
                ; --- display a box with characters ---
                ;
                        ORG  0C100H
                ;
C100 2E04               LD   L,4           ; initialize coordinate y
                ;
C102 2604       LOOPY:  LD   H,4           ; initialize coordinate x
                ;
C104 E5         LOOPX:  PUSH HL
C105 CDF303             CALL 03F3H         ; *calculate address in video-ram
C108 3623               LD   (HL),'#'      ; send a character to video-ram
C10A E1                 POP  HL
                ;
C10B 24                 INC  H             ; increment coordinate x
C10C 7C                 LD   A,H           ;
C10D FE09               CP   9             ; check coordinate x end
C10F 20F3               JR   NZ,LOOPX      ;
                ;
C111 2C                 INC  L             ; increment coordinate y
C112 7D                 LD   A,L           ;
C113 FE09               CP   9             ; check coordinate y end
C115 20EB               JR   NZ,LOOPY      ;
                ;
C117 C3665C             JP   5C66H         ; jump to system monitor
                ;
C11A                    END


*GC100

*
   #####
   #####
   #####
   #####
   #####
```

０４０１Ｈ　論理座標▷実座標変換

アドレス　０４０１Ｈ

機　能　論理座標▷実座標変換

レジスタ　ＡＦ－－－－ＨＬ

解　説　レジスタＨに格納された１～８０の横座標，およびレジスタＬに格納された１～２５の縦座標を，８０／７２／４０／３６桁モード，カラー／白黒モードを考慮した上で，ＢＡＳＩＣで使用するＶＲＡＭ(Video－ＲＡＭ)上の実座標に変換し，結果をレジスタＨおよびレジスタＬに格納してもどります．

横座標の計算は，次式に準じて行います．

カラー・モード８０桁の場合……Ｈ← (Ｈ－１)＋１
カラー・モード７２桁の場合……Ｈ← (Ｈ－１)＋２
カラー・モード４０桁の場合……Ｈ← (Ｈ－１)×２＋２
カラー・モード３６桁の場合……Ｈ← (Ｈ－１)×２＋４
白黒　　モード８０桁の場合……Ｈ← (Ｈ－１)
白黒　　モード７２桁の場合……Ｈ← (Ｈ－１)×２
白黒　　モード４０桁の場合……Ｈ← (Ｈ－１)×２
白黒　　モード３６桁の場合……Ｈ← (Ｈ－１)×２＋４

なお，８０／７２／４０／３６桁モードの判別に当たってはワーク・エリア中ＥＡ６５Ｈ番地の値を，カラー／白黒モードの判別に当たってはＥＡ６１Ｈ番地の値を，それぞれ参照します．

サンプル

```
                  ;
                  ; --- calculate physical position ---
                  ;
                       ORG   0C100H
                  ;
C100 2172C1            LD    HL,PRMPTX
C103 CD69C1            CALL  DSPMSG
C106 CD32C1            CALL  INDEC              ; input logical position x
C109 57                LD    D,A
                  ;
C10A 218CC1            LD    HL,PRMPTY
C10D CD69C1            CALL  DSPMSG
C110 CD32C1            CALL  INDEC              ; input logical position y
C113 5F                LD    E,A
                  ;
C114 D5                PUSH  DE
C115 E1                POP   HL
C116 CD0104            CALL  0401H              ;*calculate physical position
C119 E5                PUSH  HL
C11A D1                POP   DE
                  ;
C11B 21A6C1            LD    HL,ANSWRX
```

```
C11E CD69C1          CALL DSPMSG
C121 7A              LD   A,D
C122 CD4EC1          CALL DSPDEC            ; display physical position x
              ;
C125 21C0C1          LD   HL,ANSWRY
C128 CD69C1          CALL DSPMSG
C12B 7B              LD   A,E
C12C CD4EC1          CALL DSPDEC            ; display physical position y
              ;
C12F C3665C          JP   5C66H             ; jump to system monitor
              ;
              ; input decimal number subroutine
              ;
              ;    outputs  : input number in accumulator
              ;    destroys : register a,f,e
              ;
C132 1E00     INDEC: LD   E,0               ; initialize input number
C134 3E30            LD   A,'0'             ; initialize input character
              ;
C136 F5       INLOOP:PUSH AF                ;
C137 7B              LD   A,E               ;
C138 87              ADD  A,A               ;
C139 87              ADD  A,A               ;
C13A 83              ADD  A,E               ;
C13B 87              ADD  A,A               ; e <-- e * 10 + input number
C13C 5F              LD   E,A               ;
C13D F1              POP  AF                ;
C13E D630            SUB  '0'               ;
C140 83              ADD  A,E               ;
C141 5F              LD   E,A               ;
              ;
C142 CD750F          CALL 0F75H             ; input a character from keyboard
C145 CD5702          CALL 0257H             ; echo-back a character
              ;
C148 FE0D            CP   0DH
C14A 20EA            JR   NZ,INLOOP
              ;
C14C 7B              LD   A,E
C14D C9              RET
              ;
              ; display decimal number subroutine
              ;
              ;    inputs   : binary code less than 100 in accumulator
              ;    destroys : register a,f
              ;
C14E 08       DSPDEC:EX   AF,AF'
C14F AF              XOR  A
C150 08              EX   AF,AF'
C151 D60A     DSPDE1:SUB  10
C153 3805            JR   C,DSPDE2
C155 08              EX   AF,AF'
C156 3C              INC  A
C157 08              EX   AF,AF'
C158 18F7            JR   DSPDE1
              ;
C15A 08       DSPDE2:EX   AF,AF'
C15B C630            ADD  A,'0'
C15D FE30            CP   '0'
C15F C45702          CALL NZ,0257H
              ;
C162 08              EX   AF,AF'
C163 C63A            ADD  A,+10+'0'
C165 CD5702          CALL 0257H
              ;
C168 C9              RET
              ;
              ; display message subroutine
              ;
C169 7E       DSPMSG:LD   A,(HL)
C16A A7              AND  A
C16B C8              RET  Z
C16C CD5702          CALL 0257H             ; display a character
C16F 23              INC  HL
C170 18F7            JR   DSPMSG
```

```
                  ;
                  ; message area
                  ;
C172 0D0A6C6F PRMPTX:DEFB 0DH,0AH,'logical pos x (1-80) ? ',0
C176 67696361
C17A 6C20706F
C17E 73207820
C182 28312D38
C186 3029203F
C18A 2000
C18C 0D0A6C6F PRMPTY:DEFB 0DH,0AH,'logical pos y (1-25) ? ',0
C190 67696361
C194 6C20706F
C198 73207920
C19C 28312D32
C1A0 3529203F
C1A4 2000
C1A6 0D0A2070 ANSWRX:DEFB 0DH,0AH,' physical position x ? ',0
C1AA 68797369
C1AE 63616C20
C1B2 706F7369
C1B6 74696F6E
C1BA 2078203F
C1BE 2000
C1C0 0D0A2070 ANSWRY:DEFB 0DH,0AH,' physical position y ? ',0
C1C4 68797369
C1C8 63616C20
C1CC 706F7369
C1D0 74696F6E
C1D4 2079203F
C1D8 2000
                  ;
C1DA                     END
```

```
GC100

 logical pos x (1-80) ? 10
 logical pos y (1-25) ? 10
  physical position x ? 22
  physical position y ? 9
 *GC100

 logical pos x (1-80) ? 36
 logical pos y (1-25) ? 25
  physical position x ? 74
  physical position y ? 24
 *GC100

 logical pos x (1-80) ? 0
 logical pos y (1-25) ? 0
  physical position x ? 3
  physical position y ? I5
 *
```

０４５１Ｈ　テキスト・スクリーンの１行クリア

アドレス	０４５１Ｈ
機　能	テキスト・スクリーンの１行クリア
レジスタ	ＡＦＢＣＤＥＨＬ

解　説　レジスタＤＥによってＶＲＡＭ(Video－ＲＡＭ)上の先頭アドレスを指定する，ＣＲＴテキスト・スクリーンの指定ラインをクリアします．

実際には，ＶＲＡＭエリアの最後ＦＥＢ８Ｈ番地から１２０バイトの内容を，レジスタＤＥによって指定するアドレス以降にブロック転送するため，ヌル・キャラクタ・コードおよびベース・カラーは，ＦＥＢ８Ｈ番地からの１２０バイトに用意されたクリア・バッファの内容を参照します．

サンプル

```
                 ;
                 ; --- demonstration of erace lines ---
                 ;
                         ORG  0C100H
                 ;
                 ; section of phase 1
                 ;
C100 2E19                LD   L,25             ; appoint coordinate y
C102 2650        LOOPY:  LD   H,80             ; appoint coordinate x
                 ;
C104 3E7E        LOOPX:  LD   A,'~'            ; appoint a character
                 ;
C106 E5                  PUSH HL
C107 CDD702              CALL 02D7H            ; move cursor and display
C10A E1                  POP  HL
                 ;
C10B CD36C1              CALL WAITS            ; wait a moment
                 ;
C10E 25                  DEC  H                ; decrement coordinate x
C10F 20F3                JR   NZ,LOOPX
                 ;
C111 2D                  DEC  L                ; decrement coordinate y
C112 20EE                JR   NZ,LOOPY
                 ;
                 ; section of phase 2
                 ;
C114 21F0F3              LD   HL,0F300H+120+120
C117 3E08                LD   A,8
                 ;
C119 54          LOOP:   LD   D,H
C11A 5D                  LD   E,L
                 ;
C11B E5                  PUSH HL
C11C CD5104              CALL 0451H            ;*erace a line
C11F E1                  POP  HL
                 ;
C120 CD2DC1              CALL WAITL            ; wait
                 ;
C123 116801              LD   DE,120+120+120
C126 19                  ADD  HL,DE
C127 3D                  DEC  A
C128 20EF                JR   NZ,LOOP
                 ;
C12A C3665C              JP   5C66H            ; jump to system monitor
```

```
                ;
                ; wait subroutine
                ;
                ;    destroys : register b,c
                ;
C12D 0EFF       WAITL: LD   C,0FFH
C12F CD36C1     WAITL1:CALL WAITS
C132 0D                DEC  C
C133 20FA              JR   NZ,WAITL1
C135 C9                RET
                ;
                ; wait a moment subroutine
                ;
                ;    destroys : register b only
                ;
C136 06FF       WAITS: LD   B,0FFH
C138 10FE       WAITS1:DJNZ WAITS1
C13A C9                RET
                ;
C13B                   END


*GC100~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

０４５ＡＨ　テキスト・スクリーンのクリア

アドレス	０４５ＡＨ
機　　能	**テキスト・スクリーンのクリア**
レジスタ	ＡＦＢＣＤＥＨＬ

解　　説　ＣＲＴテキスト・スクリーンのみをクリアします．

　Ｆ３００Ｈ番地から３０００バイトの全ＶＲＡＭ(Video－ＲＡＭ)エリアに関し，テキスト・エリアはヌル・キャラクタ・コードによって，アトリビュート・エリアはヌル・アトリビュート・コードによって，それぞれ満たします．

　ヌル・キャラクタ・コードおよびベース・カラーは，ＦＥＢ８Ｈ番地からの１２０バイトに用意されたクリア・バッファの内容を参照します．

サンプル

```
                ;
                ; --- dump memory by ascii character ---
                ;
                        ORG   0C100H
                ;
C100 DD210000           LD    IX,0000H          ; appoint top address of memory
                ;
C104 CD5A04     LOOP:   CALL  045AH             ;*clear text screen all
                ;
C107 3A5DEA             LD    A,(0EA5DH)        ; lines of scrolling area
C10A 6F                 LD    L,A               ; set coordinate y
                ;
C10B 3A65EA     LOOPY:  LD    A,(0EA65H)        ; width of text screen
C10E 67                 LD    H,A               ; set coordinate x
                ;
C10F DB09       LOOPX:  IN    A,(09H)           ;
C111 CB47               BIT   0,A               ; jump to system monitor if stop key
C113 CA665C             JP    Z,5C66H           ;
                ;
C116 DD7E00             LD    A,(IX)            ; get a character from memory
C119 DD23               INC   IX                ; next address
                ;
C11B FE20               CP    20H               ;
C11D 3002               JR    NC,NOMAL          ; chage control character
C11F 3E2E               LD    A,'.'             ;
                ;
C121 CD5702     NOMAL:  CALL  0257H             ; diaplay a character
                ;
C124 25                 DEC   H                 ; decrement coordinate x
C125 20E8               JR    NZ,LOOPX
                ;
C127 2D                 DEC   L                 ; decrement coordinate y
C128 20E1               JR    NZ,LOOPY
                ;
C12A 18D8               JR    LOOP              ; now repeat
                ;
C12C                    END
```

```
*GC100
n call.Overflow.Out of memory.Undefined
line number.Subscript out of range.Redim
ensioned array.Division by zero.Illegal
direct.Type mismatch.Out of string space
.String too long.String formula too comp
lex.Can't continue.Undefined user functi
on.No RESUME.RESUME without error.Unprin
table error.Missing operand.Line buffer
overflow.Position not on screen.Bad File
 Data.Disk BASIC Feature.Communications
Buffer Overflow.Port not initialized.Tap
e read ERROR.モ.ノヨ.o¦".gx".G>.ノ...5Jハ¬9.v
r゜ーウr.ンGrSム¬¬..ノ reシヘr ヨw>rRヌO_ロ.ノ·D·D·D·D
·D·D·D·D·D·D. ♥ス3. ......P.. . ...(゜....
@..............y................auto ..
.........go to ..........list ..........
.run............time$...........key ...
.........print ..........list...........
.cont............./キノくノくノテ..テ.........＿.
N/M/M/..ホ/ヘ/ヘ/.........P8....〒年  .年 in .
Ok ...Break.!..9~# ＿タN#F#◣i`z
*
```

０４７ＡＨ　ライン数からＶＲＡＭのバイト数を計算

アドレス	０４７ＡＨ
機　　能	ライン数からＶＲＡＭのバイト数を計算
レジスタ	ＡＦＢＣ－－ＨＬ

解　　説　アキュムレータのデータ（１～２５５）に１２０を乗じたものを，レジスタＢＣおよびレジスタＨＬに格納します．

ただし，アキュムレータのデータが０（ゼロ）である場合のチェックは行っていないため，０は２５６とみなした上で計算を行います．

インタプリタでは，ＣＲＴテキスト・スクリーンのライン数をＶＲＡＭ（Video－ＲＡＭ）上でのバイト数に変換する際に，このシステム・サブルーチンを用いています．

サンプル

```
                ;
                ; --- multiply by 120 ---
                ;
                        ORG  0C100H
                ;
C100 2132C1     START:  LD   HL,PROMPT
C103 CD29C1             CALL DSPMSG
C106 CD7E1B             CALL 1B7EH          ; screen editor
C109 DA665C             JP   C,5C66H        ; system monitor if stop key
C10C 23                 INC  HL
                ;
C10D E5                 PUSH HL
C10E CD29C1             CALL DSPMSG         ; display the input number again
C111 214AC1             LD   HL,ANSWER
C114 CD29C1             CALL DSPMSG
C117 E1                 POP  HL
                ;
C118 CDBE2B             CALL 2BBEH          ; convert into binary code
C11B CD7F27             CALL 277FH          ; convert into integer
                ;
C11E 3AA8F0             LD   A,(0F0A8H)     ; get lsb byte of integer
C121 CD7A04             CALL 047AH          ;*multiply by 120, result in reg-hl
                ;
C124 CD132D             CALL 2D13H          ; display unsigned integer
                ;
C127 18D7               JR   START
                ;
                ; display message subroutine
                ;
C129 7E         DSPMSG:LD   A,(HL)
C12A A7                 AND  A
C12B C8                 RET  Z
C12C CD5702             CALL 0257H          ; display a character
C12F 23                 INC  HL
C130 18F7               JR   DSPMSG
                ;
                ; message area
                ;
C132 0D0A0A61   PROMPT:DEFB 0DH,0AH,0AH,'a integer in byte ? ',0
C136 20696E74
C13A 65676572
```

```
C13E 20696E20
C142 62797465
C146 203F2000
C14A 206D756C ANSWER:DEFB ' multiplied by 120 is ',0
C14E 7469706C
C152 69656420
C156 62792031
C15A 32302069
C15E 732000
                ;
C161                   END


*GC100


a integer in byte ? 1
1 multiplied by 120 is 120

a integer in byte ? 2
2 multiplied by 120 is 240

a integer in byte ? 3
3 multiplied by 120 is 360

a integer in byte ? 10
10 multiplied by 120 is 1200

a integer in byte ? 100
100 multiplied by 120 is 12000

a integer in byte ? 0
0 muitiplied by 120 is 30720

a integer in byte ?

*
```

０４８７H　テキスト画面クリア・バッファの設定

アドレス	０４８７H
機　能	テキスト画面クリア・バッファの設定
レジスタ	AFBCDEHL

解　説　VRAM(Video－RAM)エリアの直後,すなわちFEB８H番地からの１２０バイトに用意されるCRTテキスト・スクリーンのライン・クリア・バッファを，適切なデータでイニシアライズします．

インタプリタがテキスト・スクリーンをクリアする際には，このクリア・バッファの内容を，ライン単位でブロック転送して行きます．

サンプル

```
                  ;
                  ; --- apply screen clearing buffer ---
                  ;
                          ORG   0C100H
                  ;
C100 CD2E0B               CALL  0B2EH              ;
C103 AF                   XOR   A                  ; function keys display off
C104 3260EA               LD    (0EA60H),A         ;
                  ;
C107 08                   EX    AF,AF'            ; set null character in accumulator'
                  ;
C108 DB09         LOOP:   IN    A,(09H)            ;
C10A CB47                 BIT   0,A                ; jump to system monitor if stop key
C10C CA665C               JP    Z,5C66H            ;
                  ;
C10F 08                   EX    AF,AF'
C110 325AEA               LD    (0EA5AH),A         ; set null character
C113 3C                   INC   A                  ; increment null character code
C114 08                   EX    AF,AF'
                  ;
C115 CD8704               CALL  0487H              ;*initialize clearing buffer
C118 CD5A04               CALL  045AH              ; clear text screen all
C11B CD20C1               CALL  WAIT               ; wait a moment
                  ;
C11E 18E8                 JR    LOOP               ; and repeat
                  ;
                  ; wait a moment subroutine
                  ;
                  ;    destroys : register b,c
                  ;
C120 0EFF         WAIT:   LD    C,0FFH
C122 06FF         WAIT1:  LD    B,0FFH
C124 10FE         WAIT2:  DJNZ  WAIT2
C126 0D                   DEC   C
C127 20F9                 JR    NZ,WAIT1
C129 C9                   RET
                  ;
C12A                      END
```

```
*GC100HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
*HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
```

０４Ｆ８Ｈ　アトリビュートのコントロール

アドレス	０４Ｆ８Ｈ
機　　能	アトリビュートのコントロール
レジスタ	ＡＦＢＣＤＥＨＬ

解　説　レジスタＨに格納された１～８０の横座標，およびレジスタＬに格納された１～２５の縦座標，によって指定するテキスト・スクリーン上の指定座標に対応するアトリビュートを，アキュムレータに格納されたアトリビュート・コードによって設定します．

なお，アキュムレータに与えるアトリビュート・コードは，μＰＤ３３０１(プログラマブルＣＲＴコントローラ）に使用されているものです．

サンプル

```
                  ;
                  ; --- demonstration of control attribute ---
                  ;
                        ORG  0C100H
                  ;
                  ;                    attributes code map
                  ;     +---------+-------------------------+-------------+
                  ;     !  bits   !          color          !     bw      !
                  ;     +---------+-------------------------+-------------+
                  ;     ! bit0=0  ! **unused** ! normal     ! normal      !
                  ;     !     =1  ! **unused** ! secret     ! secret      !
                  ;     ! bit1=0  ! **unused** ! normal     ! normal      !
                  ;     !     =1  ! **unused** ! blinking   ! blinking    !
                  ;     ! bit2=0  ! **unused** ! normal     ! normal      !
                  ;     !     =1  ! **unused** ! reverse    ! reverse     !
                  ;     +---------+-------------------------+-------------+
                  ;     ! bit3=0  !            appoint mode ! **unused**  !
                  ;     !     =1  ! appoint color           ! **unused**  !
                  ;     +---------+-------------------------+-------------+
                  ;     ! bit4=0  ! character  ! normal     ! normal      !
                  ;     !     =1  ! graphic    ! over line  ! over line   !
                  ;     ! bit5=0  ! blue off   ! normal     ! normal      !
                  ;     !     =1  ! blue on    ! under line ! under line  !
                  ;     ! bit6=0  ! red off    ! **unused** ! **unused**  !
                  ;     !     =1  ! red on     ! **unused** ! **unused**  !
                  ;     ! bit7=0  ! green off  ! **unused** ! character   !
                  ;     !     =1  ! green on   ! **unused** ! graphic     !
                  ;     +---------+-------------------------+-------------+
                  ;
C100 01FF00             LD   BC,00FFH         ;
C103 CDF708             CALL 08F7H            ;
C106 0628               LD   B,40             ; console ,,0,1:width 40,25
C108 0E19               LD   C,25             ;
C10A CD3A09             CALL 093AH            ;
                  ;
C10D 2E01               LD   L,1              ; appoint coordinate y
                  ;
C10F 2601         LOOPY: LD  H,1              ; appoint coordinate x
                  ;
C111 78           LOOPX: LD  A,B              ;
C112 3C                 INC  A                ;
C113 E607               AND  07H              ;
C115 47                 LD   B,A              ;
```

```
C116 CB17               RL    A            ; generate attribute code
C118 CB17               RL    A            ;   from basic function code
C11A CB17               RL    A            ;
C11C CB17               RL    A            ;
C11E CB17               RL    A            ;
C120 F618               OR    18H          ;
                ;
C122 F5                 PUSH  AF           ;
C123 C5                 PUSH  BC           ;
C124 E5                 PUSH  HL           ;
C125 CDF804             CALL  04F8H        ;*set attribute
C128 E1                 POP   HL           ;
C129 C1                 POP   BC           ;
C12A F1                 POP   AF           ;
                ;
C12B 24                 INC   H            ; increment coordinate x
                ;
C12C F5                 PUSH  AF           ;
C12D C5                 PUSH  BC           ;
C12E E5                 PUSH  HL           ;
C12F CDF804             CALL  04F8H        ;*set attribute
C132 E1                 POP   HL           ;
C133 C1                 POP   BC           ;
C134 F1                 POP   AF           ;
                ;
C135 24                 INC   H            ; increment coordinate x
                ;
C136 F5                 PUSH  AF           ;
C137 C5                 PUSH  BC           ;
C138 E5                 PUSH  HL           ;
C139 CDF804             CALL  04F8H        ;*set attribute
C13C E1                 POP   HL           ;
C13D C1                 POP   BC           ;
C13E F1                 POP   AF           ;
                ;
C13F 24                 INC   H            ; increment coordinate x
                ;
C140 7C                 LD    A,H          ;
C141 FE28               CP    39+1         ; check coordinate x end or not
C143 20CC               JR    NZ,LOOPX     ;
                ;
C145 2C                 INC   L            ; increment coordinate y
                ;
C146 7D                 LD    A,L          ;
C147 FE1A               CP    25+1         ; check coordinate y end or not
C149 20C4               JR    NZ,LOOPY     ;
                ;
C14B C3665C             JP    5C66H        ; jump to system monitor
                ;
C14E                    END


*GC100
*
```

０６６４H　ライン番号▷ＶＲＡＭアドレス変換

アドレス	0664H
機　能	ライン番号▷ＶＲＡＭアドレス変換
レジスタ	AF−−DEHL

解　説　アキュムレータに格納された０～２５の縦座標に対応するＶＲＡＭ（Video－ＲＡＭ）の先頭アドレスを，レジスタＤＥに求めます．

アキュムレータに与えられた縦座標が０の場合には，ＶＲＡＭの開始アドレスであるＦ３００Ｈが結果となりますが，縦座標が２５の場合にはテキスト・スクリーン・クリア・バッファの開始アドレスであるＦＥＢ８Ｈが結果になります．

また，このシステム・サブルーチンでは，ライン番号からＶＲＡＭアドレスへの変換を計算によって行っているのではなく，高速性を重視するため，０６７５Ｈ～０６Ａ８Ｈ番地のテーブルよりアドレスを引き出しています．

サンプル

```
                ;
                ; --- appoint the line on text screen ---
                ;
                        ORG   0C100H
                ;
C100 215FC1     START:  LD    HL,PRMPT1
C103 CD56C1             CALL  DSPMSG
C106 CD2FC1             CALL  INDEC             ; input line number in accumulator
C109 FE19               CP    25
C10B D2665C             JP    NC,5C66H
                ;
C10E CD6406             CALL  0664H             ;*get address on vram in reg-de
                ;
C111 217DC1             LD    HL,PRMPT2
C114 CD56C1             CALL  DSPMSG
C117 CD750F             CALL  0F75H             ; input a character in accumulator
C11A CD5702             CALL  0257H             ; echo-back the character
                ;
                ; flash the appointed character ten times
                ;
C11D 060A               LD    B,10
                ;
C11F 12         FLASH:  LD    (DE),A
C120 CD4BC1             CALL  WAIT
                ;
C123 F5                 PUSH  AF
C124 3E20               LD    A,' '
C126 12                 LD    (DE),A
C127 CD4BC1             CALL  WAIT
C12A F1                 POP   AF
                ;
C12B 10F2               DJNZ  FLASH
                ;
C12D 18D1               JR    START
                ;
                ; input decimal number subroutine
                ;
                ;    outputs  : input number in accumulator
```

```
                ;    destroys : register a,f,e
                ;
C12F 1E00       INDEC: LD   E,0                ; initialize input number
C131 3E30              LD   A,'0'              ; initialize input character
                ;
C133 F5         INLOOP:PUSH AF                 ;
C134 7B                LD   A,E                ;
C135 87                ADD  A,A                ;
C136 87                ADD  A,A                ;
C137 83                ADD  A,E                ;
C138 87                ADD  A,A                ; e <-- e * 10 + input number
C139 5F                LD   E,A                ;
C13A F1                POP  AF                 ;
C13B D630              SUB  '0'                ;
C13D 83                ADD  A,E                ;
C13E 5F                LD   E,A                ;
                ;
C13F CD750F            CALL 0F75H              ; input a character from keyboard
C142 CD5702            CALL 0257H              ; echo-back a character
                ;
C145 FE0D              CP   0DH
C147 20EA              JR   NZ,INLOOP
                ;
C149 7B                LD   A,E
C14A C9                RET
                ;
                ; wait subroutine
                ;
                ;    destroys : register h,l
                ;
C14B 2EFF       WAIT:  LD   L,0FFH
C14D 26FF       WAIT1: LD   H,0FFH
C14F 25         WAIT2: DEC  H
C150 20FD              JR   NZ,WAIT2
C152 2D                DEC  L
C153 20F8              JR   NZ,WAIT1
C155 C9                RET
                ;
                ; display message subroutine
                ;
C156 7E         DSPMSG:LD   A,(HL)
C157 A7                AND  A
C158 C8                RET  Z
C159 CD5702            CALL 0257H              ; display a character
C15C 23                INC  HL
C15D 18F7              JR   DSPMSG
                ;
                ; message area
                ;
C15F 0D0A6C69   PRMPT1:DEFB 0DH,0AH,'line number from 0 to 24 ? ',0
C163 6E65206E
C167 756D6265
C16B 72206672
C16F 6F6D2030
C173 20746F20
C177 3234203F
C17B 2000
C17D 0D0A2020   PRMPT2:DEFB 0DH,0AH,'              character ? ',0
C181 20202020
C185 20202020
C189 20202020
C18D 20636861
C191 72616374
C195 6572203F
C199 2000
                ;
C19B                   END


*GC100

line number from 0 to 24 ? 20
              character ?
*
```

０６Ｅ６Ｈ　アトリビュートのコントロール

アドレス	０６Ｅ６Ｈ
機　能	アトリビュートのコントロール
レジスタ	ＡＦＢＣＤＥＨＬ

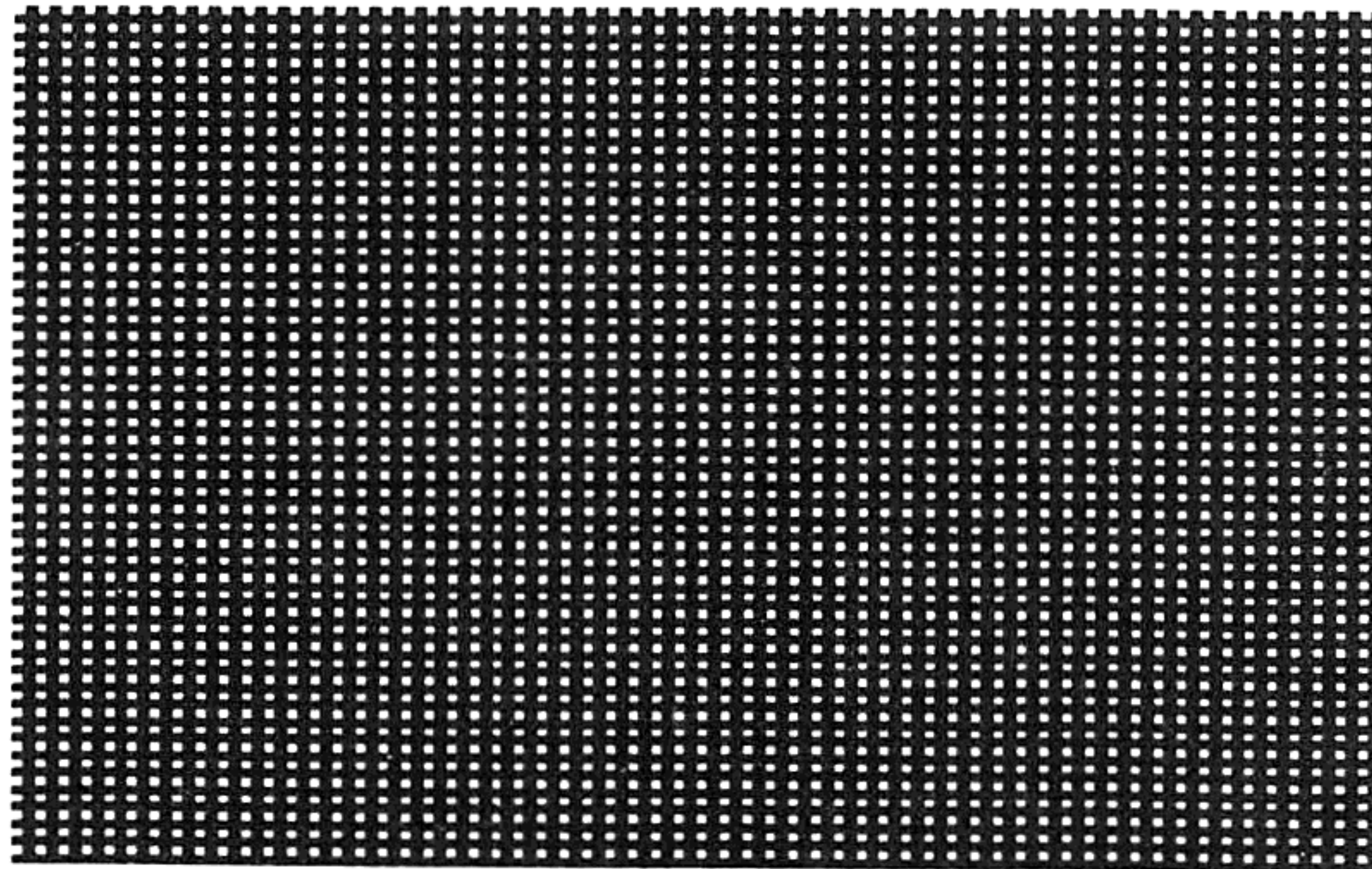
解　説　ＥＤＣ４Ｈ番地に格納された１～８０の横座標，およびＥＤＣ３Ｈ番地に格納された１～２５の縦座標，によって指定するＣＲＴスクリーン上の指定座標に対応するアトリビュートを，ＥＤＢＡＨ番地に格納されたアトリビュート・コードによって設定します．

なお，ＥＤＢＡＨ番地に与えるアトリビュート・コードは，μＰＤ３３０１（プログラマブルＣＲＴコントローラ）に使用されているものです．

サンプル

```
                  ;
                  ; --- demonstration of control attribute ii ---
                  ;
                          ORG   0C100H
                  ;
                  ;                      attributes code map
                  ;       +---------+------------------------+------------+
                  ;       !  bits   !          color         !     bw     !
                  ;       +---------+------------------------+------------+
                  ;       ! bit0=0  ! **unused** ! normal      ! normal     !
                  ;       !     =1  ! **unused** ! secret      ! secret     !
                  ;       ! bit1=0  ! **unused** ! normal      ! normal     !
                  ;       !     =1  ! **unused** ! blinking    ! blinking   !
                  ;       ! bit2=0  ! **unused** ! normal      ! normal     !
                  ;       !     =1  ! **unused** ! reverse     ! reverse    !
                  ;       +---------+------------+-----------+------------+
                  ;       ! bit3=0  !              appoint mode ! **unused** !
                  ;       !     =1  ! appoint color             ! **unused** !
                  ;       +---------+------------+-----------+------------+
                  ;       ! bit4=0  ! character  ! normal      ! normal     !
                  ;       !     =1  ! graphic    ! over line   ! over line  !
                  ;       ! bit5=0  ! blue off   ! normal      ! normal     !
                  ;       !     =1  ! blue on    ! under line  ! under line !
                  ;       ! bit6=0  ! red off    ! **unused**  ! **unused** !
                  ;       !     =1  ! red on     ! **unused**  ! **unused** !
                  ;       ! bit7=0  ! green off  ! **unused**  ! character  !
                  ;       !     =1  ! green on   ! **unused**  ! graphic    !
                  ;       +---------+------------+-----------+------------+
                  ;
C100 010000               LD    BC,0000H        ;
C103 CDF708               CALL  08F7H           ;
C106 0650                 LD    B,80            ; console ,,0,0:width 80,25
C108 0E19                 LD    C,25            ;
C10A CD3A09               CALL  093AH           ;
                  ;
C10D 2E01                 LD    L,1             ; appoint coordinate y
                  ;
C10F 2601         LOOPY:  LD    H,1             ; appoint coordinate x
                  ;
C111 22C3ED       LOOPX:  LD    (0EDC3H),HL     ; store coordinate x and y
C114 3E80                 LD    A,80H           ; attribute code of semi-graphic
C116 32BAED               LD    (0EDBAH),A      ; store it
                  ;
C119 F5                   PUSH  AF
C11A C5                   PUSH  BC
```

```
C124 E5                 PUSH HL
C125 CDF303             CALL 03F3H          ; calculate vram address
C128 36F5               LD   (HL),0F5H      ; send the form in semi-grphic
C12A E1                 POP  HL
C12B C1                 POP  BC
C12C F1                 POP  AF
                ;
C12D 24                 INC  H              ; increment coordinate x
                ;
C12E 7C                 LD   A,H            ;
C12F FE51               CP   80+1           ; check coordinate x end or not
C131 20DE               JR   NZ,LOOPX       ;
                ;
C133 2C                 INC  L              ; increment coordinate y
                ;
C134 7D                 LD   A,L            ;
C135 FE1A               CP   25+1           ; check coordinate y end or not
C137 20D6               JR   NZ,LOOPY       ;
                ;
C139 C3665C             JP   5C66H          ; jump to system monitor
                ;
C13C                    END


*GC100
*
```

07C9H　ファンクション・キー表示の開始

アドレス	07C9H
機　　能	ファンクション・キー表示の開始
レジスタ	AFBCDEHL

解　　説　アキュムレータの内容に1を加えた番号の，プログラマブル・ファンクション・キーから始まる5個のプログラマブル・ファンクション・キーを，テキスト・スクリーン最下段の定位置に表示します．

ただし，あらかじめファンクション・キー表示モードが指定されていることが必要です．

サンプル

```
                ;
                ; --- control function keys display ---
                ;
                        ORG   0C100H
                ;
C100 214DC1     START:  LD    HL,PROMPT
C103 CD44C1             CALL  DSPMSG
                ;
C106 CD750F             CALL  0F75H            ; input a character from keyboard
C109 CD5702             CALL  0257H            ; echo-back the character
                ;
C10C FE03               CP    03H
C10E CA665C             JP    Z,5C66H
                ;
C111 FE79               CP    'y'
C113 280E               JR    Z,YES
C115 FE59               CP    'Y'
C117 280A               JR    Z,YES
C119 FE6E               CP    'n'
C11B 281E               JR    Z,NO
C11D FE4E               CP    'N'
C11F 281A               JR    Z,NO
                ;
C121 18DD               JR    START
                ;
C123 3EFF       YES:    LD    A,0FFH           ;
C125 3260EA             LD    (0EA60H),A       ;
C128 2167C1             LD    HL,NUMBER        ;
C12B CD44C1             CALL  DSPMSG           ;
C12E CD750F             CALL  0F75H            ; display appointed function keys
C131 CD5702             CALL  0257H            ;
C134 D631               SUB   '1'              ;
C136 CDC907             CALL  07C9H            ;
C139 18C5               JR    START            ;
                ;
C13B AF         NO:     XOR   A                ;
C13C 3260EA             LD    (0EA60H),A       ; not display function keys
C13F CD2E0B             CALL  0B2EH            ;
C142 18BC               JR    START
                ;
                ; display message subroutine
                ;
C144 7E         DSPMSG:LD     A,(HL)
C145 A7                 AND   A
C146 C8                 RET   Z
```

```
C147 CD5702          CALL 0257H           ; display a character
C14A 23              INC  HL
C14B 18F7            JR   DSPMSG
              ;
              ; message area
              ;
C14D 0D0A6469 PROMPT:DEFB 0DH,0AH,'display function key ? ',0
C151 73706C61
C155 79206675
C159 6E637469
C15D 6F6E206B
C161 6579203F
C165 2000
C167 0D0A2064 NUMBER:DEFB 0DH,0AH,' display from number f·',0
C16B 6973706C
C16F 61792066
C173 726F6D20
C177 6E756D62
C17B 65722066
C17F A500
              ;
C181                 END


*GC100

display function key ? n
display function key ? y
 display from number f·1
display function key ? y
 display from number f·2
display function key ? y
 display from number f·3
display function key ? y
 display from number f·4
display function key ? y
 display from number f·5
display function key ? y
 display from number f·6
display function key ?
*
```

０８Ｆ７Ｈ　テキスト・スクリーンのモード設定

アドレス	０８Ｆ７Ｈ
機　　能	テキスト・スクリーンのモード設定
レジスタ	ＡＦＢＣＤＥ－－

解　　説　以下に示す各レジスタに与える入力パラメータにより，ＣＲＴのテキストスクリーンに関するモード設定を行います．

レジスタＢ……ファンクション・キー表示を行う場合にはＦＦＨを，ファンクション・キー表示を行わない場合には００Ｈを与えます．

レジスタＣ……カラー・モードの場合にはＦＦＨを，白黒モードの場合には００Ｈを与えます．

サンプル

```
               ;
               ; --- change text screen mode ---
               ;
                      ORG   0C100H
               ;
               ; display prompt message and get 1st parameter
               ;
C100 2166C1           LD    HL,CONSOL
C103 CD5DC1           CALL  DSPMSG
C106 CD3AC1           CALL  INDEC
C109 1C               INC   E
C10A 7B               LD    A,E
C10B 325EEA           LD    (0EA5EH),A       ; set number of scroll start line
               ;
               ; get 2nd parameter
               ;
C10E CD58C1           CALL  DSPCOM
C111 CD3AC1           CALL  INDEC
C114 7B               LD    A,E
C115 325DEA           LD    (0EA5DH),A       ; set lines of scroll area
               ;
               ; get 3rd parameter
               ;
C118 CD58C1           CALL  DSPCOM
C11B CD3AC1           CALL  INDEC
C11E AF               XOR   A
C11F 1C               INC   E
C120 1D               DEC   E
C121 2802             JR    Z,NO
C123 3EFF             LD    A,0FFH
C125 47        NO:    LD    B,A              ; set display function keys or not
               ;
               ; get 4th parameter
               ;
C126 CD58C1           CALL  DSPCOM
C129 CD3AC1           CALL  INDEC
C12C AF               XOR   A
C12D 1C               INC   E
C12E 1D               DEC   E
C12F 2802             JR    Z,BW
C131 3EFF             LD    A,0FFH
```

```
C133 4F        BW:     LD   C,A              ; set color mode or b/w mode
               ;
               ; main works of change text screen mode
               ;
C134 CDF708            CALL 08F7H            ;*change text screen mode
               ;
C137 C3665C            JP   5C66H            ; jump to system monitor
               ;
               ; input a decimal number subroutine
               ;
               ;    outputs  : binary code less than 256 in register e
               ;    destroys : register a,f,e
               ;
C13A 1E00      INDEC: LD    E,0              ; initialize a input number
C13C 3E30             LD    A,'0'            ; initialize a input character
               ;
C13E F5        INDEC1:PUSH AF                ;
C13F 7B               LD   A,E               ;
C140 87               ADD  A,A               ;
C141 87               ADD  A,A               ;
C142 83               ADD  A,E               ;
C143 87               ADD  A,A               ; e <-- e * 10 + input number
C144 5F               LD   E,A               ;
C145 F1               POP  AF                ;
C146 D630             SUB  '0'               ;
C148 83               ADD  A,E               ;
C149 5F               LD   E,A               ;
               ;
C14A CD750F           CALL 0F75H             ; input a character from keyboard
               ;
C14D FE3A             CP   '9'+1
C14F D0               RET  NC
C150 FE30             CP   '0'
C152 D8               RET  C
               ;
C153 CD5702           CALL 0257H             ; echo-back a character
C156 18E6             JR   INDEC1
               ;
               ; display a comma subroutine
               ;
               ;    destroys : accumulator
               ;
C158 3E2C      DSPCOM:LD   A,','
C15A C35702           JP   0257H             ; display a character
               ;
               ; display message subroutine
               ;
               ;    destroys : accumulator
               ;
C15D 7E        DSPMSG:LD   A,(HL)
C15E A7               AND  A
C15F C8               RET  Z
C160 CD5702           CALL 0257H             ; display a character
C163 23               INC  HL
C164 18F7             JR   DSPMSG
               ;
               ; message area
               ;
C166 0D0A636F  CONSOL:DEFB 0DH,0AH,'console ',0
C16A 6E736F6C
C16E 652000
               ;
C171                  END
```

```
*GC100

console 0,25,0,1
*GC100

console 10,10,1,0
*
```

０９３ＡＨ　テキスト・スクリーン表示文字数の設定

アドレス	093AH
機　能	テキスト・スクリーン表示文字数の設定
レジスタ	ＡＦＢＣＤＥ－－

解　説　テキスト・スクリーンに表示される最大の文字数を，レジスタＢに格納された桁数およびレジスタＣに格納された行数に設定します．

通常の場合，桁数としては８０／７２／４０／３６を指定し，行数としては２５／２０を指定しますが，ＢＡＳＩＣでは“Illegal function call” エラーが発生するような組み合わせ，例えば１０桁×１０行等も指定することが可能です．

サンプル

```
                ;
                ; ---  change text screen mode and width ---
                ;
                        ORG   0C100H
                ;
                ; input the 1st parameter of width
                ;
C100 2176C1             LD    HL,WIDTH
C103 CDED52             CALL  52EDH             ; display the message of width
C106 CD58C1             CALL  INDEC
C109 43                 LD    B,E               ; set width x in register b
                ;
                ; input the 2nd parameter of width
                ;
C10A 3E2C               LD    A,','
C10C CD5702             CALL  0257H
C10F CD58C1             CALL  INDEC
C112 4B                 LD    C,E               ; set width y in register c
C113 C5                 PUSH  BC                ; save parameters of width
                ;
                ; input the 1st parameter of console
                ;
C114 2181C1             LD    HL,CONSOL
C117 CDED52             CALL  52EDH             ; display the message of console
C11A CD58C1             CALL  INDEC
C11D 1C                 INC   E
C11E 7B                 LD    A,E
C11F 325EEA             LD    (0EA5EH),A        ; set number of scroll start line
                ;
                ; input the 2nd parameter of console
                ;
C122 3E2C               LD    A,','
C124 CD5702             CALL  0257H
C127 CD58C1             CALL  INDEC
C12A 7B                 LD    A,E
C12B 325DEA             LD    (0EA5DH),A        ; set lines of scroll area
                ;
                ; input the 3rd parameter of console
                ;
C12E 3E2C               LD    A,','
C130 CD5702             CALL  0257H
C133 CD58C1             CALL  INDEC
C136 AF                 XOR   A
C137 1C                 INC   E
C138 1D                 DEC   E
C139 2802               JR    Z,NO
C13B 3EFF               LD    A,0FFH
```

```
C13D 47         NO:    LD   B,A              ; set display function keys or not
                ;
                ; input the 4th parameter of console
                ;
C13E 3E2C              LD   A,','
C140 CD5702            CALL 0257H
C143 CD58C1            CALL INDEC
C146 AF                XOR  A
C147 1C                INC  E
C148 1D                DEC  E
C149 2802              JR   Z,BW
C14B 3EFF              LD   A,0FFH
C14D 4F         BW:    LD   C,A              ; set color mode or b/w mode
                ;
                ; just change text screen mode and width
                ;
C14E CDF708            CALL 08F7H            ; change text screen mode
                ;
C151 C1                POP  BC               ; resave parameters of width
C152 CD3A09            CALL 093AH            ;*change text screen width
                ;
C155 C3665C            JP   5C66H            ; jump to system monitor
                ;
                ; input a decimal number subroutine
                ;
                ;    outputs  : binary code less than 256 in register e
                ;    destroys : register a,f,e
                ;
C158 1E00       INDEC: LD   E,0              ; initialize a input number
C15A 3E30              LD   A,'0'            ; initialize a input character
                ;
C15C F5         INDEC1:PUSH AF               ;
C15D 7B                LD   A,E              ;
C15E 87                ADD  A,A              ;
C15F 87                ADD  A,A              ;
C160 83                ADD  A,E              ;
C161 87                ADD  A,A              ; reg-e <-- reg-e * 10 + the number
C162 5F                LD   E,A              ;
C163 F1                POP  AF               ;
C164 D630              SUB  '0'              ;
C166 83                ADD  A,E              ;
C167 5F                LD   E,A              ;
                ;
C168 CD750F            CALL 0F75H            ; input a charcter from keyboard
                ;
C16B FE3A              CP   '9'+1
C16D D0                RET  NC
C16E FE30              CP   '0'
C170 D8                RET  C
                ;
C171 CD5702            CALL 0257H            ; echo-back the character
C174 18E6              JR   INDEC1
                ;
                ; message area
                ;
C176 0D0A7769 WIDTH: DEFB 0DH,0AH,'width   ',0
C17A 64746820
C17E 202000
C181 0D0A636F CONSOL:DEFB 0DH,0AH,'console ',0
C185 6E736F6C
C189 652000
                ;
C18C                   END


*GC100

width   10,25
console 0,25,0,0
*
```

０９Ａ３Ｈ　テキスト・スクリーン表示桁数の設定

アドレス	０９Ａ３Ｈ
機　能	テキスト・スクリーン表示桁数の設定
レジスタ	ＡＦＢＣＤＥＨＬ

解　説　テキスト・スクリーンの１ライン中に表示する最大の桁数を，アキュムレータに格納された桁数に設定します．

通常の場合は，８０／７２／４０／３６桁を指定しますが，ＢＡＳＩＣでは“Illegal function call”エラーが発生するような設定も行うことができます．

サンプル

```
               ;
               ; --- change width x of text screen ---
               ;
                    ORG   0C100H
               ;
               ; get 1st parameter of width
               ;
C100 213FC1         LD    HL,WIDTH
C103 CD36C1         CALL  DSPMSG
               ;
C106 CD13C1         CALL  INDEC           ; input a parameter in register e
C109 CD31C1         CALL  DSPCOM          ; display a comma
C10C 7B             LD    A,E             ; set width x in accumulator
               ;
               ; just change width x of text screen
               ;
C10D CDA309         CALL  09A3H           ;*change width x of text screen
               ;
C110 C3665C         JP    5C66H           ; jump to system monitor
               ;
               ; input a decimal number subroutine
               ;
               ;    outputs  : binary code less than 256 in register e
               ;    destroys : register a,f,e
               ;
C113 1E00      INDEC: LD  E,0             ; initialize a input number
C115 3E30           LD    A,'0'           ; initialize a input character
               ;
C117 F5        INDEC1:PUSH AF             ;
C118 7B             LD    A,E             ;
C119 87             ADD   A,A             ;
C11A 87             ADD   A,A             ;
C11B 83             ADD   A,E             ;
C11C 87             ADD   A,A             ; e <-- e * 10 + input number
C11D 5F             LD    E,A             ;
C11E F1             POP   AF              ;
C11F D630           SUB   '0'             ;
C121 83             ADD   A,E             ;
C122 5F             LD    E,A             ;
               ;
C123 CD750F         CALL  0F75H           ; input a character from keyboard
               ;
C126 FE3A           CP    '9'+1
C128 D0             RET   NC
C129 FE30           CP    '0'
```

```
C12B D8                 RET   C
                ;
C12C CD5702             CALL  0257H          ; echo-back a character
C12F 18E6               JR    INDEC1
                ;
                ; display a comma subroutine
                ;
                ;    destroys : accumulator
                ;
C131 3E2C       DSPCOM:LD     A,',' 
C133 C35702             JP    0257H          ; display a character
                ;
                ; display message subroutine
                ;
                ;    destroys : accumulator
                ;
C136 7E         DSPMSG:LD     A,(HL)
C137 A7                 AND   A
C138 C8                 RET   Z
C139 CD5702             CALL  0257H          ; display a character
C13C 23                 INC   HL
C13D 18F7               JR    DSPMSG
                ;
                ; message area
                ;
C13F 0D0A7769 WIDTH: DEFB 0DH,0AH,'width ',0
C143 64746820
C147 00
                ;
C148                    END


*GC100

width 20,
*
Ok
print string$(255,"X
")
XXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXX
Ok
```

０９Ｄ７Ｈ　テキスト・スクリーン表示行数の設定

アドレス　０９Ｄ７Ｈ

機　　能　テキスト・スクリーン表示行数の設定

レジスタ　ＡＦＢＣＤＥＨＬ

解　　説　テキスト・スクリーンの１画面中に表示する最大の行数を，アキュムレータに格納された行数に設定します．

通常の場合は，２５行／２０行を指定しますが，ＢＡＳＩＣでは “Illegal function call” エラーが発生するような設定も行うことが可能です．

サンプル

```
                ;
                ; --- change width y of text screen ---
                ;
                        ORG  0C100H
                ;
                ; get 2st parameter of width
                ;
C100 213FC1             LD   HL,WIDTH
C103 CD36C1             CALL DSPMSG
                ;
C106 CD31C1             CALL DSPCOM             ; display a comma
C109 CD13C1             CALL INDEC              ; input a parameter in register e
C10C 7B                 LD   A,E                ; set width y in accumulator
                ;
                ; just change width y of text screen
                ;
C10D CDD709             CALL 09D7H              ;*change width y of text screen
                ;
C110 C3665C             JP   5C66H              ; jump to system monitor
                ;
                ; input a decimal number subroutine
                ;
                ;    outputs  : binary code less than 256 in register e
                ;    destroys : register a,f,e
                ;
C113 1E00       INDEC:  LD   E,0                ; initialize a input number
C115 3E30               LD   A,'0'              ; initialize a input character
                ;
C117 F5         INDEC1:PUSH AF                  ;
C118 7B                 LD   A,E                ;
C119 87                 ADD  A,A                ;
C11A 87                 ADD  A,A                ;
C11B 83                 ADD  A,E                ;
C11C 87                 ADD  A,A                ; e <-- e * 10 + input number
C11D 5F                 LD   E,A                ;
C11E F1                 POP  AF                 ;
C11F D630               SUB  '0'                ;
C121 83                 ADD  A,E                ;
C122 5F                 LD   E,A                ;
                ;
C123 CD750F             CALL 0F75H              ; input a character from keyboard
                ;
C126 FE3A               CP   '9'+1
C128 D0                 RET  NC
C129 FE30               CP   '0'
C12B D8                 RET  C
                ;
```

```
C12C CD5702            CALL 0257H           ; echo-back a character
C12F 18E6              JR   INDEC1
              ;
              ; display a comma subroutine
              ;
              ;    destroys : accumulator
              ;
C131 3E2C     DSPCOM:LD   A,',' 
C133 C35702            JP   0257H           ; display a character
              ;
              ; display message subroutine
              ;
              ;    destroys : accumulator
              ;
C136 7E       DSPMSG:LD   A,(HL)
C137 A7                AND  A
C138 C8                RET  Z
C139 CD5702            CALL 0257H           ; display a character
C13C 23                INC  HL
C13D 18F7              JR   DSPMSG
              ;
              ; message area
              ;
C13F 0D0A7769 WIDTH: DEFB 0DH,0AH,'width ',0
C143 64746820
C147 00
              ;
C148                   END


*GC100

width ,25
*
```

０９Ｆ６Ｈ　ＣＲＴの２５行モード用イニシアライズ

アドレス　０９Ｆ６Ｈ

機　能　ＣＲＴの２５行モード用イニシアライズ

レジスタ　ＡＦ－－－－ＨＬ

解　説　ＤＭＡコントローラ（μＰＤ８２５７）およびＣＲＴコントローラ（μＰＤ３３０１）を２５行モード用にイニシアライズします．

イニシアライズ時には，０Ａ６８Ｈ～０Ａ６ＣＨ番地のデータ・テーブルを使用します．

サンプル

```
                 ;
                 ; --- 25 or 20 lines on text screen ---
                 ;
                         ORG   0C100H
                 ;
C100 DB09        LOOP:   IN    A,(09H)        ;
C102 CB47                BIT   0,A            ; jump to system monitor if stop key
C104 CA665C              JP    Z,5C66H        ;
                 ;
C107 2140C1              LD    HL,MSG25
C10A CD37C1              CALL  DSPMSG
C10D CDF609              CALL  09F6H          ;*initialize for 25 lines mode
C110 CD28C1              CALL  WAIT
                 ;
C113 DB09                IN    A,(09H)        ;
C115 CB47                BIT   0,A            ; jump to system monitor if stop key
C117 CA665C              JP    Z,5C66H        ;
                 ;
C11A 215EC1              LD    HL,MSG20
C11D CD37C1              CALL  DSPMSG
C120 CD5E0A              CALL  0A5EH          ;*initialize for 20 lines mode
C123 CD28C1              CALL  WAIT
                 ;
C126 18D8                JR    LOOP
                 ;
                 ; wait subroutine
                 ;
                 ;     destroys : register a,f,b,c
                 ;
C128 3E0F        WAIT:   LD    A,0FH
C12A 0EFF        WAIT1:  LD    C,0FFH
C12C 06FF        WAIT2:  LD    B,0FFH
C12E 10FE        WAIT3:  DJNZ  WAIT3
C130 0D                  DEC   C
C131 20F9                JR    NZ,WAIT2
C133 3D                  DEC   A
C134 20F4                JR    NZ,WAIT1
C136 C9                  RET
                 ;
                 ; display message subroutine
                 ;
                 ;     destroys : accumulator
                 ;
C137 7E          DSPMSG:LD     A,(HL)
C138 A7                  AND   A
C139 C8                  RET   Z
C13A CD5702              CALL  0257H          ; display a character
C13D 23                  INC   HL
C13E 18F7                JR    DSPMSG
```

```
                  ;
                  ; message area
                  ;
C140 0D0A3235 MSG25: DEFB 0DH,0AH,´25 lines par text screen !!´,0
C144 206C696E
C148 65732070
C14C 61722074
C150 65787420
C154 73637265
C158 656E2021
C15C 2100
C15E 0D0A3230 MSG20: DEFB 0DH,0AH,´20 lines par text screen !!´,0
C162 206C696E
C166 65732070
C16A 61722074
C16E 65787420
C172 73637265
C176 656E2021
C17A 2100
                  ;
C17C                     END


*GC100

25 lines par text screen !!
20 lines par text screen !!
25 lines par text screen !!
20 lines par text screen !!
25 lines par text screen !!
20 lines par text screen !!
25 lines par text screen !!
20 lines par text screen !!
25 lines par text screen !!
20 lines par text screen !!
25 lines par text screen !!
20 lines par text screen !!
25 lines par text screen !!
20 lines par text screen !!
25 lines par text screen !!
*
```

０Ａ５ＥＨ　ＣＲＴの２０行モード用イニシアライズ

アドレス	０Ａ５ＥＨ
機　　能	ＣＲＴの２０行モード用イニシアライズ
レジスタ	ＡＦ－－－－ＨＬ

解　　説　ＤＭＡコントローラ（μＰＤ８２５７）およびＣＲＴコントローラ（μＰＤ３３０１）を２０行モード用にイニシアライズします．

イニシアライズ時には，０Ａ６ＤＨ～０Ａ７１Ｈ番地のデータ・テーブルを使用します．

サンプル

```
                ;
                ; --- initialize crt and dma controller ---
                ;
                        ORG   0C100H
                ;
C100 212EC1     START:  LD    HL,PROMPT
C103 CD25C1             CALL  DSPMSG
                ;
C106 CD750F             CALL  0F75H             ; input a character from keyboard
C109 CD5702             CALL  0257H             ; echo-back the character
                ;
C10C FE03               CP    03H
C10E CA665C             JP    Z,5C66H           ; jump to system monitor if stop key
                ;
C111 FE31               CP    '1'
C113 2806               JR    Z,INIT25
C115 FE32               CP    '2'
C117 2807               JR    Z,INIT20
                ;
C119 18E5               JR    START
                ;
C11B CDF609     INIT25:CALL 09F6H               ;*initialize for 25characters mode
C11E 18E0               JR    START
                ;
C120 CD5E0A     INIT20:CALL 0A5EH               ;*initialize for 20characters mode
C123 18DB               JR    START
                ;
                ; display message subroutine
                ;
                ;    destroys : accumulator
                ;
C125 7E         DSPMSG:LD   A,(HL)
C126 A7                 AND   A
C127 C8                 RET   Z
C128 CD5702             CALL  0257H             ; display a character
C12B 23                 INC   HL
C12C 18F7               JR    DSPMSG
                ;
                ; message area
                ;
C12E 0D0A3235   PROMPT:DEFB 0DH,0AH,'25 characters mode -- 1  '
C132 20636861
C136 72616374
C13A 65727320
C13E 6D6F6465
C142 202D2D20
C146 312020
```

```
C149 0D0A3230        DEFB 0DH,0AH,'20 characters mode -- 2  '
C14D 20636861
C151 72616374
C155 65727320
C159 6D6F6465
C15D 202D2D20
C161 322020
C164 0D0A2020        DEFB 0DH,0AH,'                which ? ',0
C168 20202020
C16C 20202020
C170 20202020
C174 77686963
C178 68203F20
C17C 00
               ;
C17D                 END
```

```
*GC100

25 characters mode -- 1
20 characters mode -- 2
               which ? 1
25 characters mode -- 1
20 characters mode -- 2
               which ? 2
25 characters mode -- 1
20 characters mode -- 2
               which ? 1
25 characters mode -- 1
20 characters mode -- 2
               which ? 2
25 characters mode -- 1
20 characters mode -- 2
               which ?
*
```

０Ａ７３Ｈ　ファンクション・キー・リスト

アドレス	０Ａ７３Ｈ
機　　能	ファンクション・キー・リスト
レジスタ	ＡＦＢ－ＤＥ－－

解　　説　ＢＡＳＩＣのＫＥＹＬＩＳＴと同様に，すべてのプログラマブル・ファンクション・キーの定義内容のリストを画面上に表示します．

ファンクション・キーの定義内容の一部はテキスト画面の最下段に表示させておくことができますが，このシステム・サブルーチンでは，そのすべての内容を画面に表示させることができます．

サンプル

```
                 ;
                 ; --- function key listings ---
                 ;
                        ORG   0C100H
                 ;
C100 211CC1      START: LD    HL,PROMPT
C103 CDED52             CALL  52EDH              ; display the message
                 ;
C106 CD750F             CALL  0F75H              ; input a character from keyboard
C109 CD5702             CALL  0257H              ; echo-back the character
                 ;
C10C FE79               CP    'y'
C10E 2807               JR    Z,YES
C110 FE59               CP    'Y'
C112 2803               JR    Z,YES
C114 C3665C             JP    5C66H              ; jump to system monitor if not yes
                 ;
C117 CD730A      YES:   CALL  0A73H              ;*display function key listings
C11A 18E4               JR    START
                 ;
                 ; message area
                 ;
C11C 0D0A6675    PROMPT:DEFB  0DH,0AH,'function key listings ? ',0
C120 6E637469
C124 6F6E206B
C128 6579206C
C12C 69737469
C130 6E677320
C134 3F2000
                 ;
C137                    END


*GC100

function key listings ? y
files           load˙
mount           save˙
remove          name˙
list            kill˙
run         run˙

function key listings ? n
*
```

N－BASICの主要データ領域

0675H～06A8H	テキスト・スクリーン先頭アドレス・データ
10E5H～117CH	キーボード・マトリクス変換データ
1838H～1874H	N－BASICのタイトル・メッセージ
1FC9H～1FCEH	"Bad CRLF " メッセージ
1FFFH～1FFBH	"Found：" および "Skip：" メッセージ
24F2H～24F5H	単精度実数型の1
24F6H～25F2H	log演算用のデータ
2B26H～2B2DH	単精度実数型の100および10
30D4H～3147H	単精度実数型および倍精度実数型のデータ群
3148H～3151H	16進数から10進数への変換データ
3233H～3253H	exp演算用のデータ
32EAH～32F5H	乱数発生用のデータ
3340H～3347H	三角関数演算用のデータ
3348H～335CH	sin演算用のデータ
3398H～33BCH	atn演算用のデータ
33BDH～3466H	中間言語から処理ルーチンへの分岐テーブル
3467H～34C0H	関数中間言語から処理ルーチンへの分岐テーブル
34C1H～34F4H	頭文字とキーワード格納アドレスのインデクス
34F5H～37A7H	キーワードと中間言語の変換テーブル
37A8H～37BCH	特殊記号と中間言語の変換テーブル
37C9H～37F0H	型変換および演算ルーチンへの分岐テーブル
37F1H～3A04H	エラー・メッセージ・テーブル
3A05H～3B5AH	ワーク・エリアの初期データ
3B5BH～3B5FH	" in "メッセージ
3B60H～3B65H	"Ok CRLF " メッセージ
3B66H～3B6BH	"Break" メッセージ
48ACH～48BEH	" ?Redo from start CRLF " メッセージ
49D7H～49E7H	" ?Extra ignored CRLF " メッセージ
5BF0H～5BFFH	"Undefined line " メッセージ
5C6FH～5C92H	機械語モニタのコマンド・テーブル

０Ｂ１８Ｈ　ブート・ストラップ・ローダ

アドレス	０Ｂ１８Ｈ
機　　能	ブート・ストラップ・ローダ
レジスタ	ＡＦＢＣＤＥＨＬ

解　　説　ミニ・フロッピ・ディスク・ユニット（ＰＣ－８０３１，ＰＣ８０３１－２Ｗ，ＰＣ－８０Ｓ３１相当）の，ドライブ１，トラック１，セクタ１から２５６バイトのデータを読み出し，Ｃ０００Ｈ～Ｃ０ＦＦＨ番地に格納後，エラーが発生しなければＣ０００Ｈ番地にジャンプします．

エラーが発生した場合には，ＥＤＣ７Ｈ番地に００Ｈを格納するのみでもどります．

このシステム・サブルーチンは，ＢＡＳＩＣがコールド・スタートする際に，ディスク・ユニットが接続されていれば必ずコールします．

サンプル

```
                  ;
                  ; --- boot strap loader ---
                  ;
                         ORG   0C100H
                  ;
C100 2147C1       LABEL1:LD    HL,PROMPT
C103 CD3EC1              CALL  DSPMSG
                  ;
C106 CD750F              CALL  0F75H            ; input a character from keyboard
C109 CD5702              CALL  0257H            ; echo-back the character
                  ;
C10C FE03                CP    03H
C10E CA665C              JP    Z,5C66H
                  ;
C111 FE79                CP    'y'
C113 2804                JR    Z,LABEL2
C115 FE59                CP    'Y'
C117 20E7                JR    NZ,LABEL1
                  ;
C119 2166C1       LABEL2:LD    HL,SURE
C11C CD3EC1              CALL  DSPMSG
                  ;
C11F CD750F              CALL  0F75H            ; input a character from keyboard
C122 CD5702              CALL  0257H            ; echo-back the character
                  ;
C125 FE03                CP    03H
C127 CA665C              JP    Z,5C66H
                  ;
C12A FE79                CP    'y'
C12C 2804                JR    Z,LABEL3
C12E FE59                CP    'Y'
C130 20CE                JR    NZ,LABEL1
                  ;
C132 CD180B       LABEL3:CALL  0B18H            ;*boot strap loader
                  ;
C135 2185C1              LD    HL,ERROR         ;
C138 CD3EC1              CALL  DSPMSG           ; diaplay error message and exit
```

```
C13B C3665C          JP    5C66H                 ;
                ;
                ; display message subroutine
                ;
                ;    inputs   : top address of message in register hl
                ;    destroys : register a,f,h,l
                ;
C13E 7E         DSPMSG:LD    A,(HL)
C13F A7                AND   A
C140 C8                RET   Z
C141 CD5702            CALL  0257H               ; display a character
C144 23                INC   HL
C145 18F7              JR    DSPMSG
                ;
                ; message area
                ;
C147 0D0A       PROMPT:DEFB 0DH,0AH
C149 626F6F74          DEFB 'boot the disk-basic again ? ',0
C14D 20746865
C151 20646973
C155 6B2D6261
C159 73696320
C15D 61676169
C161 6E203F20
C165 00
C166 0D0A       SURE:  DEFB 0DH,0AH
C168 6974206D          DEFB 'it means cold start, sure ? ',0
C16C 65616E73
C170 20636F6C
C174 64207374
C178 6172742C
C17C 20737572
C180 65203F20
C184 00
C185 0D0A       ERROR: DEFB 0DH,0AH
C187 63616E27          DEFB 'can''t load the IPL !!',07H,0
C18B 74206C6F
C18F 61642074
C193 68652049
C197 504C2021
C19B 210700
                ;
C19E                   END


*GC100

boot the disk-basic again ? y
it means cold start, sure ? n
boot the disk-basic again ? y
it means cold start, sure ? n
boot the disk-basic again ? y
can't load the IPL !!
*
```

０Ｂ２ＥＨ　テキスト・スクリーン最下段の消去

アドレス	０Ｂ２ＥＨ
機　能	テキスト・スクリーン最下段の消去
レジスタ	ＡＦＢＣＤＥＨＬ

解　説　ＣＲＴテキスト・スクリーンの最下段を消去します．

ＢＡＳＩＣでは，このシステム・サブルーチンを，ファンクション・キー表示エリアを消去するために使用していますが，実際にファンクション・キー表示を行うか否かは，ワーク・エリアＥＡ６０Ｈ番地の内容が００ＨかＦＦＨかによって決定します．

サンプル

```
                    ;
                    ; --- blink the function key display---
                    ;
                           ORG   0C100H
                    ;
C100 3EFF                  LD    A,0FFH
C102 3260EA                LD    (0EA60H),A        ; turn on the function key display
                    ;
C105 DB09           BLINK: IN    A,(09H)           ;
C107 CB47                  BIT   0,A               ; jump to system monitor if stop key
C109 CA665C                JP    Z,5C66H           ;
                    ;
C10C 2142C1                LD    HL,MSGDSF
C10F CD39C1                CALL  DSPMSG
C112 CDC907                CALL  07C9H             ; display function key
C115 CD2DC1                CALL  WAIT
                    ;
C118 DB09                  IN    A,(09H)           ;
C11A CB47                  BIT   0,A               ; jump to system monitor if stop key
C11C CA665C                JP    Z,5C66H           ;
                    ;
C11F 215CC1                LD    HL,MSGERA
C122 CD39C1                CALL  DSPMSG
C125 CD2E0B                CALL  0B2EH             ;*erace the last line on screen
C128 CD2DC1                CALL  WAIT
                    ;
C12B 18D8                  JR    BLINK
                    ;
                    ; wait subroutine
                    ;
                    ;     destroys : flags only
                    ;
C12D C5             WAIT:  PUSH  BC
C12E 0EFF                  LD    C,0FFH
C130 06FF           WAIT1: LD    B,0FFH
C132 10FE           WAIT2: DJNZ  WAIT2
C134 0D                    DEC   C
C135 20F9                  JR    NZ,WAIT1
C137 C1                    POP   BC
C138 C9                    RET
                    ;
                    ; display message subroutine
                    ;
                    ;     inputs   : top address of message in register hl
```

```
                ;    destroys : register a,f,h,l
                ;
C139 7E         DSPMSG:LD   A,(HL)
C13A A7                AND  A
C13B C8                RET  Z
C13C CD5702            CALL 0257H           ; display a character
C13F 23                INC  HL
C140 18F7              JR   DSPMSG
                ;
                ; message area
                ;
C142 0D0A6469 MSGDSP:DEFB 0DH,0AH,'display function key !!',0
C146 73706C61
C14A 79206675
C14E 6E637469
C152 6F6E206B
C156 65792021
C15A 2100
C15C 0D0A6572 MSGERA:DEFB 0DH,0AH,'erase the last line !!',0
C160 61736520
C164 74686520
C168 6C617374
C16C 206C696E
C170 65202121
C174 00
                ;
C175                   END


*GC100

display function key !!
erase the last line !!
display function key !!
erase the last line !!
display function key !!
erase the last line !!
display function key !!
erase the last line !!
display function key !!
erase the last line !!
display function key !!
erase the last line !!
display function key !!
erase the last line !!
display function key !!
erase the last line !!
display function key !!
erase the last line !!
display function key !!
erase the last line !!
*
```

０Ｂ９２Ｈ　グラフィック座標▷キャラクタ座標変換

アドレス　０Ｂ９２Ｈ

機　　能　グラフィック座標▷キャラクタ座標変換

レジスタ　ＡＦ－－－－ＨＬ

解　　説　レジスタＨに格納された１～１６０のセミ・グラフィック横座標およびレジスタＬに格納された１～１００のセミ・グラフィック縦座標を，キャラクタ横座標およびキャラクタ縦座標に変換します．

サンプル

```
                   ;
                   ; --- from '2001 : a space odyssey' ---
                   ;
                           ORG   0C100H
                   ;
C100 01FF00                LD    BC,00FFH          ;
C103 CDF708                CALL  08F7H             ;
C106 0650                  LD    B,80              ; console ,,0,1:width 80,25
C108 0E19                  LD    C,25              ;
C10A CD3A09                CALL  093AH             ;
                   ;
C10D 2100F8                LD    HL,0F800H         ;
C110 225AEA                LD    (0EA5AH),HL       ; color 7,0,1:print chr$(12);
C113 CD5A04                CALL  045AH             ;
                   ;
C116 3E01                  LD    A,1               ;
C118 32E1C2                LD    (CHECK),A         ;
C11B 110A00                LD    DE,10             ;
C11E DD21E5C2              LD    IX,DATA           ;
C122 0632                  LD    B,50              ; initialize work area
C124 CD42C2        MAIN1:  CALL  MAKED             ;
C127 DD19                  ADD   IX,DE             ;
C129 05                    DEC   B                 ;
C12A 20F8                  JR    NZ,MAIN1          ;
                   ;
C12C 0601                  LD    B,1
C12E 0E3C          MAIN2:  LD    C,60
C130 C5            MAIN4:  PUSH  BC
C131 CD76C1                CALL  STARS
C134 C1                    POP   BC
C135 CD36C2                CALL  WAIT
C138 0D                    DEC   C
C139 20F5                  JR    NZ,MAIN4
                   ;
C13B 04                    INC   B
C13C 78                    LD    A,B
C13D FE0F                  CP    15
C13F 20ED                  JR    NZ,MAIN2
                   ;
C141 0E0A          MAIN6:  LD    C,10
C143 C5            MAIN7:  PUSH  BC
C144 CD76C1                CALL  STARS
C147 C1                    POP   BC
C148 CD36C2                CALL  WAIT
C14B 0D                    DEC   C
C14C 20F5                  JR    NZ,MAIN7
                   ;
C14E 04                    INC   B
C14F 78                    LD    A,B
C150 FE32                  CP    50
C152 20ED                  JR    NZ,MAIN6
C154 1614                  LD    D,20
```

```
C156 1EC8       MAIN5: LD   E,200
C158 7A                LD   A,D
C159 FE02              CP   2
C15B CC71C1            CALL Z,EXIT
                ;
C15E 0632       MAIN3: LD   B,50
C160 D5                PUSH DE
C161 CD76C1            CALL STARS
C164 D1                POP  DE
C165 1D                DEC  E
C166 20F6              JR   NZ,MAIN3
C168 15                DEC  D
C169 CC71C1            CALL Z,EXIT
C16C 20E8              JR   NZ,MAIN5
C16E C3665C            JP   5C66H
                ;
C171 AF         EXIT:  XOR  A
C172 32E1C2            LD   (CHECK),A
C175 C9                RET
                ;
                ; move stars subroutine
                ;
                ;    destroys : register a,f,b,c,d,e,h,l,ix
                ;
C176 DD21E5C2 STARS: LD   IX,DATA
C17A 110A00            LD   DE,10
C17D C5         STARS1:PUSH BC
C17E D5                PUSH DE
C17F CD89C1            CALL STAR
C182 D1                POP  DE
C183 C1                POP  BC
C184 DD19              ADD  IX,DE
C186 10F5              DJNZ STARS1
C188 C9                RET
                ;
                ; move a star subroutine
                ;
                ;    destroys : register a,f,b,c,h,l
                ;
C189 DD3508     STAR:  DEC  (IX+8)
C18C C0                RET  NZ
                ;
C18D DD7E09            LD   A,(IX+9)
C190 DD7708            LD   (IX+8),A
                ;
C193 DD6E05            LD   L,(IX+5)        ; get lsb byte of last address
C196 DD6606            LD   H,(IX+6)        ; get msb byte of last address
C199 DD7E07            LD   A,(IX+7)        ; get last semi-grphic pattern code
C19C A7                AND  A
C19D CA34C2            JP   Z,LESS
                ;
C1A0 AE                XOR  (HL)
C1A1 77                LD   (HL),A          ; just clear the dot
                ;
C1A2 21A1C2            LD   HL,VECTOR       ;
C1A5 1600              LD   D,0             ;
C1A7 DD5E00            LD   E,(IX)          ;
C1AA 19                ADD  HL,DE           ;
C1AB DD7E01            LD   A,(IX+1)        ;
C1AE 86                ADD  A,(HL)          ;
C1AF 47                LD   B,A             ;
C1B0 23                INC  HL              ;
C1B1 DD7E02            LD   A,(IX+2)        ;
C1B4 86                ADD  A,(HL)          ;
C1B5 4F                LD   C,A             ;
C1B6 DD3503            DEC  (IX+3)          ; move the star
C1B9 200E              JR   NZ,SEARCH       ;
C1BB 23                INC  HL              ;
C1BC 78                LD   A,B             ;
C1BD 86                ADD  A,(HL)          ;
C1BE 47                LD   B,A             ;
C1BF 23                INC  HL              ;
C1C0 79                LD   A,C             ;
C1C1 86                ADD  A,(HL)          ;
```

```
C1C2 4F              LD   C,A              ;
C1C3 DD7E04          LD   A,(IX+4)         ;
C1C6 DD7703          LD   (IX+3),A         ;
                ;
C1C9 78         SEARCH:LD   A,B            ;
C1CA FE01            CP   1                ;
C1CC 3822            JR   C,NEW            ; check x on screen or not
C1CE FE9C            CP   156              ;
C1D0 301E            JR   NC,NEW           ;
                ;
C1D2 79              LD   A,C              ;
C1D3 FE01            CP   1                ;
C1D5 3819            JR   C,NEW            ; check y on screen or not
C1D7 FE62            CP   98               ;
C1D9 3015            JR   NC,NEW           ;
                ;
C1DB DD7001     PSET:   LD   (IX+1),B      ; store coordinate x
C1DE DD7102          LD   (IX+2),C         ; store coordinate y
                ;
C1E1 CD02C2          CALL CAL              ; calculate address and pattern
                ;
C1E4 DD7707          LD   (IX+7),A         ; store semi-graphic pattern code
                ;
C1E7 AE              XOR  (HL)             ; pset dot or preset dot
C1E8 77              LD   (HL),A           ; just send a dot to video-ram
                ;
C1E9 DD7505          LD   (IX+5),L         ; store lsb byte of address on vram
C1EC DD7406          LD   (IX+6),H         ; store msb byte of address on vram
C1EF C9              RET
                ;
                ; make a new star subroutine
                ;
                ;    destroys : register a,f
                ;
C1F0 3AE1C2     NEW:    LD   A,(CHECK)
C1F3 A7              AND  A
C1F4 2803            JR   Z,ERASE
C1F6 C342C2          JP   MAKED
                ;
                ; break a old star subroutine
                ;
                ;    destroys : register a,f
                ;
C1F9 DD360700   ERASE:  LD   (IX+7),0
C1FD AF              XOR  A
C1FE 32C9F8          LD   (0F8C9H),A
C201 C9              RET
                ;
                ; calculare vram address and pattern code subroutine
                ;
                ;    inputs   : coordinate x base is zero in register b
                ;               coordinate y base is zero in register c
                ;    outputs  : address on video-ram in register hl
                ;               semi-graphic pattern code in accumulator
                ;    destroys : register a,f,b,c,h,l
                ;
C202 C5         CAL:    PUSH BC
C203 60              LD   H,B
C204 69              LD   L,C
C205 24              INC  H
C206 2C              INC  L
C207 CD920B          CALL 0B92H            ;*convert into cursor position
C20A CDF303          CALL 03F3H            ; calculate address on vram
C20D C1              POP  BC
                ;
                ; calculate semi-graphic pattern code section
                ;
C20E CB40            BIT  0,B
C210 2004            JR   NZ,N1
                ;
C212 060F            LD   B,0FH
C214 1802            JR   N2
                ;
C216 06F0       N1:     LD   B,0F0H
```

```
                 ;
C218 79          N2:     LD    A,C
C219 E603                AND   03H
                 ;
C21B 3D                  DEC   A
C21C 2004                JR    NZ,N3
C21E 3E22                LD    A,22H
C220 1810                JR    N4
                 ;
C222 3D          N3:     DEC   A
C223 2004                JR    NZ,N5
C225 3E44                LD    A,44H
C227 1809                JR    N4
                 ;
C229 3D          N5:     DEC   A
C22A 2004                JR    NZ,N6
C22C 3E88                LD    A,88H
C22E 1802                JR    N4
                 ;
C230 3E11        N6:     LD    A,11H
                 ;
C232 A0          N4:     AND   B
                 ;
C233 C9                  RET
                 ;
                 ; waste fitting time subroutine
                 ;
                 ;     destroys : register a,f,b,d
                 ;
C234 0631        LESS:   LD    B,49
C236 3E32        WAIT:   LD    A,50
C238 90                  SUB   B
C239 1614        WAIT2:  LD    D,20
C23B 15          WAIT1:  DEC   D
C23C 20FD                JR    NZ,WAIT1
C23E 3D                  DEC   A
C23F 20F8                JR    NZ,WAIT2
C241 C9                  RET
                 ;
                 ; initialize work area for a new star
                 ;
                 ;     input    : top address of work area in register ix
                 ;     outputs  : work area indexed register ix
                 ;     destroys : register a,f
                 ;
C242 CD6FC2      MAKED:  CALL  RND1
C245 DD7700              LD    (IX+0),A
C248 DD360150            LD    (IX+1),80
C24C DD360232            LD    (IX+2),50
C250 CD79C2              CALL  RND2
C253 DD7703              LD    (IX+3),A
C256 DD7704              LD    (IX+4),A
C259 DD3605C9            LD    (IX+5),0C9H
C25D DD3606F8            LD    (IX+6),0F8H
C261 DD360704            LD    (IX+7),04H
C265 CD80C2              CALL  RND3
C268 DD7708              LD    (IX+8),A
C26B DD7709              LD    (IX+9),A
C26E C9                  RET
                 ;
                 ; generate a random number multiply from 1 to 15 by 4 subroutine
                 ;
                 ;     outputs  : generated number in accumulator
                 ;     destroys : register a,f
                 ;
C26F CD87C2      RND1:   CALL  RSUB
C272 E60F                AND   15
C274 CB07                RLC   A
C276 CB07                RLC   A
C278 C9                  RET
                 ;
                 ; generate a random number from 1 to 8 subroutine
                 ;
                 ;     outputs  : generated number in accumulator
```

```
                  ;    destroys : register a,f
                  ;
C279 CD87C2       RND2:   CALL RSUB
C27C E607                 AND  7
C27E 3C                   INC  A
C27F C9                   RET
                  ;
                  ; generate a random number from 1 to 4 subroutine
                  ;
                  ;    outputs  : generated number in accumulator
                  ;    destroys : register a,f
                  ;
C280 CD87C2       RND3:   CALL RSUB
C283 E603                 AND  3
C285 3C                   INC  A
C286 C9                   RET
                  ;
                  ; generate a random number subroutine
                  ;
                  ;    outputs  : generated number in accumulator
                  ;    destroys : register a,f
                  ;
C287 D5           RSUB:   PUSH DE
C288 E5                   PUSH HL
C289 ED5F                 LD   A,R
C28B 1600                 LD   D,0
C28D 5F                   LD   E,A
C28E 2AE3C2               LD   HL,(RNDM)
C291 19                   ADD  HL,DE
C292 22E3C2               LD   (RNDM),HL
C295 7D                   LD   A,L
C296 CB0F                 RRC  A
C298 CB0F                 RRC  A
C29A CB0F                 RRC  A
C29C CB0F                 RRC  A
C29E E1                   POP  HL
C29F D1                   POP  DE
C2A0 C9                   RET
                  ;
                  ; vector table of demonstration program
                  ;
C2A1 00FFFF00     VECTOR:DEFB 0,-1,-1,0
C2A5 00FF0100            DEFB 0,-1,1,0
C2A9 01FFFF00            DEFB 1,-1,-1,0
C2AD 01FF0001            DEFB 1,-1,0,1
C2B1 010000FF            DEFB 1,0,0,-1
C2B5 01000001            DEFB 1,0,0,1
C2B9 010100FF            DEFB 1,1,0,-1
C2BD 0101FF00            DEFB 1,1,-1,0
C2C1 00010100            DEFB 0,1,1,0
C2C5 0001FF00            DEFB 0,1,-1,0
C2C9 FF010100            DEFB -1,1,1,0
C2CD FF0100FF            DEFB -1,1,0,-1
C2D1 FF000001            DEFB -1,0,0,1
C2D5 FF0000FF            DEFB -1,0,0,-1
C2D9 FFFF0001            DEFB -1,-1,0,1
C2DD FFFF01FF            DEFB -1,-1,1,-1
                  ;
                  ; work area of demonstration program
                  ;
C2E1              CHECK: DEFS 2
C2E3              RNDM:  DEFS 2
C2E5              DATA:  END
```

μＰＤ３３０１のアトリビュート

Ｎ－ＢＡＳＩＣおよびＮ80－ＢＡＳＩＣでは，ＣＲＴテキスト・スクリーンのコントロールを実現するために，μＰＤ３３０１（プログラマブルＣＲＴコントローラ）を，文字コード・エリアとアトリビュート・エリアを分離したトランスペアレント・アトリビュート・モードで使用しています．

アトリビュート・コード表

ビット＼モード		カラーモード		白黒モード
		カラー指定	機能指定	
0	ＯＦＦ	未使用	ノーマル	ノーマル
	Ｏ Ｎ	未使用	シークレット	シークレット
1	ＯＦＦ	未使用	ノーマル	ノーマル
	Ｏ Ｎ	未使用	ブリンク	ブリンク
2	ＯＦＦ	未使用	ノーマル	ノーマル
	Ｏ Ｎ	未使用	リバース	リバース
3	ＯＦＦ	機能指定		
	Ｏ Ｎ	カラー指定		
4	ＯＦＦ	キャラクタ	ノーマル	ノーマル
	Ｏ Ｎ	グラフィック	オーバー・ライン	オーバー・ライン
5	ＯＦＦ	Ｂ信号ＯＦＦ	ノーマル	ノーマル
	Ｏ Ｎ	Ｂ信号ＯＮ	アンダー・ライン	アンダー・ライン
6	ＯＦＦ	Ｒ信号ＯＦＦ	未使用	未使用
	Ｏ Ｎ	Ｒ信号ＯＮ	未使用	未使用
7	ＯＦＦ	Ｇ信号ＯＦＦ	未使用	キャラクタ
	Ｏ Ｎ	Ｇ信号ＯＮ	未使用	グラフィック

ＲＧＢ信号▷カラー変換表

信号	黒 ブラック	青 ブルー	赤 レッド	紫 マゼンダ	緑 グリーン	水 シアン	黄 イエロー	白 ホワイト
Ｇ	ＯＦＦ	ＯＦＦ	ＯＦＦ	ＯＦＦ	Ｏ Ｎ	Ｏ Ｎ	Ｏ Ｎ	Ｏ Ｎ
Ｒ	ＯＦＦ	ＯＦＦ	Ｏ Ｎ	Ｏ Ｎ	ＯＦＦ	ＯＦＦ	Ｏ Ｎ	Ｏ Ｎ
Ｂ	ＯＦＦ	Ｏ Ｎ	ＯＦＦ	Ｏ Ｎ	ＯＦＦ	Ｏ Ｎ	ＯＦＦ	Ｏ Ｎ

０ＢＤ２Ｈ　カーソル表示の停止

アドレス	０ＢＤ２Ｈ
機　　能	カーソル表示の停止
レジスタ	ＡＦ－－－－ＨＬ

解　説　ＣＲＴスクリーン上の現カーソル位置に表示するべきカーソルを表示しません．ただし，インタプリタ等からカーソル表示ルーチンが実行された場合，ＣＲＴコントローラ（μＰＤ３３０１）のイニシアライズが行われた場合等は，カーソル表示が再開されます．

サンプル

```
                ;
                ; --- cursor on and off ---
                ;
                       ORG   0C100H
                ;
C100 DB09       LOOP:  IN    A,(09H)          ;
C102 CB47              BIT   0,A              ; jump to system monitor if stop key
C104 CA665C            JP    Z,5C66H          ;
                ;
C107 2140C1            LD    HL,MSGON
C10A CD37C1            CALL  DSPMSG
C10D CDE20B            CALL  0BE2H            ;*display cursor
C110 CD28C1            CALL  WAIT
                ;
C113 DB09              IN    A,(09H)          ;
C115 CB47              BIT   0,A              ; jump to system monitor if stop key
C117 CA665C            JP    Z,5C66H          ;
                ;
C11A 2150C1            LD    HL,MSGOFF
C11D CD37C1            CALL  DSPMSG
C120 CDD20B            CALL  0BD2H            ; not display cursor
C123 CD28C1            CALL  WAIT
                ;
C126 18D8              JR    LOOP
                ;
                ; wait subroutine
                ;
                ;    destroys : register a,f,b,c
                ;
C128 3E0F       WAIT:  LD    A,0FH
C12A 0EFF       WAIT1: LD    C,0FFH
C12C 06FF       WAIT2: LD    B,0FFH
C12E 10FE       WAIT3: DJNZ  WAIT3
C130 0D                DEC   C
C131 20F9              JR    NZ,WAIT2
C133 3D                DEC   A
C134 20F4              JR    NZ,WAIT1
C136 C9                RET
                ;
                ; display message subroutine
                ;
                ;    destroys : register a,f,h,l
                ;
```

```
C137 7E        DSPMSG:LD   A,(HL)
C138 A7               AND  A
C139 C8               RET  Z
C13A CD5702           CALL 0257H
C13D 23               INC  HL
C13E 18F7             JR   DSPMSG
               ;
               ; message area
               ;
C140 0D0A6375 MSGON: DEFB 0DH,0AH,'cursor on !! ',0
C144 72736F72
C148 206F6E20
C14C 21212000
C150 0D0A6375 MSGOFF:DEFB 0DH,0AH,'cursor off !! ',0
C154 72736F72
C158 206F6666
C15C 20212120
C160 00
               ;
C161                  END


*GC100

cursor on !! ■
cursor off !!
cursor on !! ■
cursor off !!
cursor on !! ■
cursor off !!
cursor on !! ■
cursor off !!
cursor on !! ■
cursor off !!
cursor on !! ■
cursor off !!
cursor on !! ■
cursor off !!
cursor on !! ■
cursor off !!
cursor on !! ■
cursor off !!
cursor on !! ■
*
```

０ＢＥ２Ｈ　カーソル表示の開始

アドレス	０ＢＥ２Ｈ
機　　能	カーソル表示の開始
レジスタ	ＡＦ－－－－ＨＬ

解　　説　ＣＲＴスクリーン上の，カーソル位置に表示するべきカーソルを表示します．

通常，ＢＡＳＩＣ等によって業務用のプログラム中でキーボードからのデータ入力を行う場合には，見掛け上の同時処理を実現するためおよびエラー処理の完璧を期するために，普通の入力命令（ＩＮＰＵＴ等）を避けてＩＮＫＥＹ＄等を使用しますが，ＩＮＫＥＹ＄の実行時にはカーソルが消去されています．このような場合にカーソル表示を実現するためにも，このカーソル表示ルーチンを利用することができます．

サンプル

```
100 '
110 ' input routine  for business program
120 '
130 DEFINT A-Z
140 DEF USR=&HBE2
150 BYTES=30
160 '
170 ' main routine
180 '
190 PRINT
200 PRINT "input ? ";
210 GOSUB 270
220 IF RESULT$="end" OR RESULT$="END" THEN END
230 PRINT
240 PRINT "print : ";RESULT$
250 GOTO 170
260 '
270 ' input string subroutine
280 '
290 COLUMN=0
300 RESULT$=""
310 PRINT STRING$(BYTES,"_");STRING$(BYTES,CHR$(&H1D));
320 '
330 ' main loop of input a character
340 '
350 DUMMY=USR(DUMMY)
360 IN$=INKEY$
370 IF IN$="" THEN 330
380 '
390 IN=ASC(IN$)
400 IF IN= &H8 THEN GOSUB 510:GOTO 330
410 IF IN=&H7F THEN GOSUB 510:GOTO 330
420 IF IN= &HD THEN RETURN
430 IF IN<&H20 THEN 330
440 '
```

```
450  IF COLUMN=BYTES THEN 330
460  COLUMN=COLUMN+1
470  RESULT$=RESULT$+IN$
480  PRINT IN$;
490  GOTO 330
500 '
510 ' back-spacing subroutine
520 '
530  IF COLUMN=0 THEN RETURN
540  COLUMN=COLUMN-1
550  RESULT$=LEFT$(RESULT$,COLUMN)
560  PRINT CHR$(&H1D);"_";CHR$(&H1D);
570  RETURN
```

```
run

input ? This is a dog.________________
print : This is a dog.

input ? This is my dog._______________
print : This is my dog.

input ? My dog is in the box._________
print : My dog is in the box.

input ? This is a cat.________________
print : This is a cat.

input ? This is my cat._______________
print : This is my cat.

input ? My cat is in the box._________
print : My cat is in the box.

input ? end___________________________
Ok
```

０ＢＦ３Ｈ　ＣＭＴからの入力イニシアライズ

アドレス	０ＢＦ３Ｈ
機　　能	ＣＭＴからの入力イニシアライズ
レジスタ	ＡＦ－－－－－－－

解　説　μＰＤ８２５１Ｃ（ＵＳＡＲＴ）をＣＭＴの入力用にイニシアライズ後，内蔵モータ・リレーを“ＯＮ”にしてキャリアを検出すればもどります．

ＳＴＯＰキーの押下で，“Tape read ERROR”エラーを発生します．

また，Ｆ１Ｂ６Ｈ番地へのフックが用意されています．

０Ｃ２ＥＨ　ＣＭＴインタフェースのクローズ

アドレス	０Ｃ２ＥＨ
機　　能	ＣＭＴインタフェースのクローズ
レジスタ	ＡＦ－－－－－－－

解　説　時間待ち後，μＰＤ８２５１Ｃ(USART＝Universal Synchronous Asynchronous Resiver Transmitter)をクローズし，内蔵リレーを“ＯＦＦ”にしてもどります．

０Ｃ４６Ｈ　ＣＭＴへの出力イニシアライズ

アドレス	０Ｃ４６Ｈ
機　　能	ＣＭＴへの出力イニシアライズ
レジスタ	ＡＦ－－－－－－－

解　説　μＰＤ８２５１Ｃ（ＵＳＡＲＴ）をＣＭＴの出力用にイニシアライズ後，内蔵リレーの“ＯＮ”を実行し，時間待ちを行ってもどります．

また，Ｆ１Ｂ６Ｈ番地へのフックが用意されています．

０Ｃ８８Ｈ　ＣＭＴからの１バイト入力

アドレス　０Ｃ８８Ｈ
機　　能　ＣＭＴからの１バイト入力
レジスタ　ＡＦ－－－－－－－

解　　説　ＣＭＴインタフェースから１バイトのデータを入力し，アキュムレータに格納してもどりますが，"Tape read ERROR"エラーの発生またはＳＴＯＰキーの押下でインタプリタの制御下に入ります．

また，Ｆ１Ａ４Ｈ番地へのフックが用意されています．

０ＣＢ３Ｈ　アスタリスクのフラッシング

アドレス　０ＣＢ３Ｈ
機　　能　アスタリスクのフラッシング
レジスタ　ＡＦ－－－－－－－

解　　説　テキスト・スクリーン上の定位置にアスタリスク"＊"が表示されている場合にはそれを消去し，そうでない場合には表示してもどります．

インタプリタでは，ＢＡＳＩＣのプログラムがロード中であることを通知するために，アスタリスクをフラッシングさせてします．

０ＣＤＡＨ　ＣＭＴへの１バイト出力

アドレス　０ＣＤＡＨ
機　　能　ＣＭＴへの１バイト出力
レジスタ　ＡＦ－－－－－－－

解　　説　ＣＭＴインタフェースへ，アキュムレータに格納されたデータ１バイトを出力してもどりますが，"Tape read ERROR"エラーの発生またはＳＴＯＰキーの押下でインタプリタの制御下に入ります．

また，Ｆ２０ＤＨ番地へのフックが用意されています．

0CF1H　STOPキー・チェック

アドレス	0CF1H
機　　能	STOPキー・チェック
レジスタ	AF−−−−−−

解　　説　STOP（CTRL+C）キー，ESCキー共に押下されていなければゼロ・フラグ（Z）を1にセットして，STOPキーが押下されていればキャリ・フラグ（CY）を1にセットしてもどります．

サンプル

```
                 ;
                 ; --- check stop key and esc key ---
                 ;
                        ORG   0C100H
                 ;
C100 2137C1      START: LD    HL,MESAGE
C103 CD2EC1             CALL  DSPMSG
C106 CD24C1             CALL  WAIT
                 ;
C109 CDF10C             CALL  0CF1H          ;*check stop key and esc key
C10C 28F2               JR    Z,START        ; not break and not escape
C10E 380B               JR    C,STOP         ; yes break and not escape
                 ;                           ; not break and yes escape
C110 2154C1      ESCAPE:LD    HL,MSGESC
C113 CD2EC1             CALL  DSPMSG
C116 CD750F             CALL  0F75H
C119 18E5               JR    START
                 ;
C11B 216CC1      STOP:  LD    HL,MSGSTP
C11E CD2EC1             CALL  DSPMSG
C121 C3665C             JP    5C66H          ; jump to system montiror
                 ;
                 ; wait subroutine
                 ;
                 ;     destroys : register b,c
                 ;
C124 0EFF        WAIT:  LD    C,0FFH
C126 06FF        WAIT1: LD    B,0FFH
C128 10FE        WAIT2: DJNZ  WAIT2
C12A 0D                 DEC   C
C12B 20F9               JR    NZ,WAIT1
C12D C9                 RET
                 ;
                 ; display message subroutine
                 ;
                 ;     inputs   : top address of message in register hl
                 ;     destroys : register a,f,h,l
                 ;
C12E 7E          DSPMSG:LD    A,(HL)
C12F A7                 AND   A
C130 C8                 RET   Z
C131 CD5702             CALL  0257H          ; display a character
C134 23                 INC   HL
C135 18F7               JR    DSPMSG
                 ;
                 ; message area
                 ;
C137 0D0A7075 MESAGE:DEFB 0DH,0AH,'push stop key or esc key !',0
```

```
C13B 73682073
C13F 746F7020
C143 6B657920
C147 6F722065
C14B 7363206B
C14F 65792021
C153 00
C154 20206573 MSGESC:DEFB '  esc key was pushed !!',0
C158 63206B65
C15C 79207761
C160 73207075
C164 73686564
C168 20212100
C16C 20207374 MSGSTP:DEFB '  stop key was pushed !!',0
C170 6F70206B
C174 65792077
C178 61732070
C17C 75736865
C180 64202121
C184 00
                    ;
C185                        END


*GC100

push stop key or esc key !
push stop key or esc key !
push stop key or esc key !
push stop key or esc key !
push stop key or esc key !
push stop key or esc key !
push stop key or esc key !  esc key was pushed !!■
push stop key or esc key !
push stop key or esc key !
push stop key or esc key !
push stop key or esc key !
push stop key or esc key !
push stop key or esc key !  esc key was pushed !!■
push stop key or esc key !
push stop key or esc key !
push stop key or esc key !  esc key was pushed !!■
push stop key or esc key !
push stop key or esc key !
push stop key or esc key !
push stop key or esc key !
push stop key or esc key !  esc key was pushed !!■
push stop key or esc key !  stop key was pushed !!
*
```

０Ｄ１４Ｈ　μＰＤ８２５１Ｃソフトウェア・リセット

アドレス	０Ｄ１４Ｈ
機　能	μＰＤ８２５１Ｃソフトウェア・リセット
レジスタ	Ａ－－－－－－－－

解　説　μＰＤ８２５１Ｃ(ＵＳＡＲＴ＝Universal Synchronous Asynchronous Resiver Transmitter)にリセット・コマンドを送出して，以後モード設定を受け付ける状態にします．

具体的には，μＰＤ８２５１Ｃがリセット直後である場合を考慮して，１度ダミーのデータを出力した後，実際のリセット・コマンドを送出しています．すなわち，出力ポート２１Ｈ番地に０ＥＨおよび４０Ｈを連続して出力します．

サンプル

```
                ;
                ; --- reset the usart ---
                ;
                        ORG  0C100H
                ;
C100 213EC1     START:  LD   HL,PROMPT
C103 CD35C1             CALL DSPMSG
                ;
C106 CD750F             CALL 0F75H              ; input a character from keyboard
C109 CD5702             CALL 0257H              ; echo-back the character
                ;
C10C FE03               CP   03H
C10E 281C               JR   Z,NORSET
C110 FE6E               CP   'n'
C112 2818               JR   Z,NORSET
C114 FE4E               CP   'N'
C116 2814               JR   Z,NORSET
                ;
C118 FE79               CP   'y'
C11A 2804               JR   Z,RESET
C11C FE59               CP   'Y'
C11E 20E0               JR   NZ,START
                ;
C120 CD140D     RESET:  CALL 0D14H              ;*reset uPD8251C the USART
C123 21C0C1             LD   HL,YES
C126 CD35C1             CALL DSPMSG
C129 C3665C             JP   5C66H
                ;
C12C 21C9C1     NORSET:LD    HL,NOT
C12F CD35C1             CALL DSPMSG
C132 C3665C             JP   5C66H
                ;
                ; display message subroutine
                ;
                ;    inputs   : top address of message in register hl
                ;    destroys : register a,f,h,l
                ;
C135 7E         DSPMSG:LD    A,(HL)
C136 A7                 AND  A
```

```
C137 C8              RET  Z
C138 CD5702          CALL 0257H          ; display a character
C13B 23              INC  HL
C13C 18F7            JR   DSPMSG
                ;
                ; message area
                ;
C13E 0D0A7265 PROMPT:DEFB 0DH,0AH,'reset USART (Universal'
C142 73657420
C146 55534152
C14A 54202855
C14E 6E697665
C152 7273616C
C156 0D0A2020        DEFB 0DH,0AH,'              Synchronous'
C15A 20202020
C15E 20202020
C162 20202053
C166 796E6368
C16A 726F6E6F
C16E 7573
C170 0D0A2020        DEFB 0DH,0AH,'              Asynchronous'
C174 20202020
C178 20202020
C17C 20202041
C180 73796E63
C184 68726F6E
C188 6F7573
C18B 0D0A2020        DEFB 0DH,0AH,'              Resiver'
C18F 20202020
C193 20202020
C197 20202052
C19B 65736976
C19F 6572
C1A1 0D0A2020        DEFB 0DH,0AH,'              Transmitter) ? '
C1A5 20202020
C1A9 20202020
C1AD 20202054
C1B1 72616E73
C1B5 6D697474
C1B9 65722920
C1BD 3F20
C1BF 00              DEFB 0
                ;
C1C0 0D0A7965 YES:   DEFB 0DH,0AH,'yes !!',0
C1C4 73202121
C1C8 00
C1C9 0D0A6E6F NOT:   DEFB 0DH,0AH,'not reset !!',0
C1CD 74207265
C1D1 73657420
C1D5 212100
                ;
C1D8                 END

*GC100

reset USART (Universal
             Synchronous
             Asynchronous
             Resiver
             Transmitter) ? a
reset USART (Universal
             Synchronous
             Asynchronous
             Resiver
             Transmitter) ? n
not reset !!
*GC100

reset USART (Universal
             Synchronous
             Asynchronous
             Resiver
             Transmitter) ? y
yes !!
*
```

０Ｄ４３Ｈ　一定時間の内蔵ブザー鳴動

アドレス　０Ｄ４３Ｈ
機　　能　一定時間の内蔵ブザー鳴動
レジスタ　Ａ－－－－－－－－

解　　説　ＢＡＳＩＣのＢＥＥＰと同様に一定時間（約０．５秒間）内蔵ブザーを鳴動させてもどります．すでに内蔵ブザーが鳴動中であった場合には，一定時間（約０．５秒間）内蔵ブザーの鳴動を止めてもどります．

サンプル

```
                ;
                ; --- test program of beep ---
                ;
                       ORG  0C100H
                ;
C100 0696              LD   B,150
                ;
C102 DB09       LOOP:  IN   A,(09H)             ;
C104 CB47              BIT  0,A                 ; jump to system monitor if stop key
C106 CA665C            JP   Z,5C66H             ;
                ;
C109 2120C1            LD   HL,MESAGE
C10C CD17C1            CALL DSPMSG
                ;
C10F CD430D            CALL 0D43H               ;*ring the bell for appointed time
                ;
C112 10EE              DJNZ LOOP
                ;
C114 C3665C            JP   5C66H               ; jump to system monitor if finish
                ;
                ; display message subroutine
                ;
                ;    inputs   : top address of message in register hl
                ;    destroys : register a,f,h,l
                ;
C117 7E         DSPMSG:LD   A,(HL)
C118 A7                AND  A
C119 C8                RET  Z
C11A CD5702            CALL 0257H               ; display a character
C11D 23                INC  HL
C11E 18F7              JR   DSPMSG
                ;
                ; message area
                ;
C120 62656570   MESAGE:DEFB 'beep ',0
C124 2000
                ;
C126                   END
```

```
*GC100
beep beep beep beep beep beep beep beep
beep beep beep beep beep beep beep beep
beep beep beep beep beep beep beep beep
beep beep beep beep beep beep beep beep
beep beep beep beep beep beep beep beep
beep beep beep beep beep beep beep beep
beep beep beep beep beep beep beep beep
beep beep beep beep beep beep beep beep
beep beep beep beep beep beep beep beep
beep beep beep beep beep beep beep beep
beep beep beep beep beep beep beep beep
beep beep beep beep beep beep beep beep
beep beep beep beep beep beep beep beep
beep beep beep beep beep beep beep beep
beep beep beep beep beep beep beep beep
beep beep beep beep beep beep beep beep
beep beep beep beep beep beep beep beep
beep beep beep beep beep beep beep beep
beep beep beep beep beep beep
*
```

０Ｄ４ＢＨ　内蔵ブザーのコントロール

アドレス	０Ｄ４ＢＨ
機　　能	内蔵ブザーのコントロール
レジスタ	ＡＦ－－－Ｅ－－

解　　説　アキュムレータに格納されたデータが００Ｈの場合には，内蔵ブザーをＯＦＦ（ＢＥＥＰ　０）の状態に，アキュムレータに格納されたデータが００Ｈ以外の場合には内蔵ブザーをＯＮ（ＢＥＥＰ　１）の状態に設定してもどります．

ＰＣ－８００１ｍｋⅡでは，出力ポート４０Ｈ番地の第５ビットを直接に制御することで内蔵ブザーをコントロールすることが可能ですが，この出力ポートには，ブザーの他にもいくつかの周辺機器が接続されています．したがって，この出力ポートへの出力を実行する場合には，第５ビット以外のビットが変化しないように，ＥＡ６７Ｈ番地に格納された値(以前４０Ｈ番地に出力したデータ）との間で，論理和（ＯＲ）を求めるなどの注意が必要です．

サンプル

```
                ;
                ; --- american patrol siren ---
                ;
                        ORG  0C100H
                ;
C100 160A               LD   D,10
                ;
C102 DB09       LOOP1:  IN   A,(09H)           ;
C104 CB47               BIT  0,A               ; jump to system monitor if stop key
C106 CA665C             JP   Z,5C66H           ;
                ;
C109 0E01               LD   C,1
                ;
C10B 41         LOOP2:  LD   B,C
C10C CD25C1             CALL BEEP1
C10F 10FE       THIS1:  DJNZ THIS1
                ;
C111 41                 LD   B,C
C112 CD20C1             CALL BEEP0
C115 10FE       THIS2:  DJNZ THIS2
                ;
C117 0C                 INC  C
C118 20F1               JR   NZ,LOOP2
                ;
C11A 15                 DEC  D
C11B 20E5               JR   NZ,LOOP1
                ;
C11D C3665C             JP   5C66H
                ;
                ; buzzer off subroutine
                ;
                ;    destroys : register a,f,e
                ;
C120 AF         BEEP0:  XOR  A
C121 CD4B0D             CALL 0D4BH             ;*control buzzer
C124 C9                 RET
                ;
                ; buzzer on subroutine
                ;
                ;    destroys : register a,f,e
                ;
C125 3E01       BEEP1:  LD   A,1
C127 CD4B0D             CALL 0D4BH             ;*control buzzer
C12A C9                 RET
                ;
C12B                    END


*GC100

*
```

０Ｄ６０Ｈ　プリンタへの１バイト出力

アドレス	０Ｄ６０Ｈ
機　能	プリンタへの１バイト出力
レジスタ	―――――――――

解　説　アキュムレータに格納された１バイトのデータをプリンタへ出力します.

ただし，常にＳＴＯＰキー・チェックのシステム・サブルーチンを呼び出しており，ＳＴＯＰキーの押下が認められた場合には，プリンタへの出力を止めてもどります.

また，レジスタＡＦの値をスタックに退避後，Ｆ１ＢＣＨ番地へのフックが用意されています.

サンプル

```
                ;
                ; --- output message to printer ---
                ;
                        ORG   0C100H
                ;
C100 210EC1             LD    HL,MESAGE
                ;
C103 7E         LOOP:   LD    A,(HL)          ; get a character
C104 A7                 AND   A               ; check end mark or not end mark
C105 CA665C             JP    Z,5C66H         ; system monitor if end mark
                ;
C108 CD600D             CALL  0D60H           ;*output a character to printer
C10B 23                 INC   HL              ; increment pointer
C10C 18F5               JR    LOOP
                ;
                ; message area
                ;
C10E 2D2D2054   MESAGE:DEFB '-- This is the message !! --',0DH,0
C112 68697320
C116 69732074
C11A 6865206D
C11E 65737361
C122 67652021
C126 21202D2D
C12A 0D00
                ;
C12C                    END


*GC100
-- This is the message !! --
*
```

０ＤＡ３Ｈ　内蔵リレーの反転

アドレス　０ＤＡ３Ｈ
機　能　内蔵リレーの反転
レジスタ　ＡＦ－－－－－－－

解　説　ＢＡＳＩＣのＭＯＴＯＲと同様に内蔵リレーの状態を反転します．すなわち，リレーがＯＦＦの状態であればＯＮ（ＭＯＴＯＲ　1）の状態に，リレーがＯＮの状態であればＯＦＦ（ＭＯＴＯＲ　0）の状態に，それぞれ設定してもどります．

サンプル

```
                ;
                ; --- test program of motor ---
                ;
                       ORG  0C100H
                ;
C100 0664              LD   B,100
                ;
C102 DB09       LOOP:  IN   A,(09H)           ;
C104 CB47              BIT  0,A               ; jump to system monitor if stop key
C106 CA665C            JP   Z,5C66H           ;
                ;
C109 2120C1            LD   HL,MESAGE
C10C CD17C1            CALL DSPMSG
                ;
C10F CDA30D            CALL 0DA3H             ;*reverse the motor relay
                ;
C112 10EE              DJNZ LOOP
                ;
C114 C3665C            JP   5C66H             ; jump to system monitor if finish
                ;
                ; display message subroutine
                ;
                ;    inputs   : top address of message in register hl
                ;    destroys : register a,f,h,l
                ;
C117 7E         DSPMSG:LD   A,(HL)
C118 A7                AND  A
C119 C8                RET  Z
C11A CD5702            CALL 0257H             ; display a character
C11D 23                INC  HL
C11E 18F7              JR   DSPMSG
                ;
                ; message area
                ;
C120 6D6F746F   MESAGE:DEFB 'motor   ',0
C124 72202020
C128 00
                ;
C129                   END
```

```
*GC100
motor   motor   motor   motor   motor
motor   motor   motor   motor   motor
motor   motor   motor   motor   motor
motor   motor   motor   motor   motor
motor   motor   motor   motor   motor
motor   motor   motor   motor   motor
motor   motor   motor   motor   motor
motor   motor   motor   motor   motor
motor   motor   motor   motor   motor
motor   motor   motor   motor   motor
motor   motor   motor   motor   motor
motor   motor   motor   motor   motor
motor   motor   motor   motor   motor
motor   motor   motor   motor   motor
motor   motor   motor   motor   motor
motor   motor   motor   motor   motor
motor   motor   motor   motor   motor
motor   motor   motor   motor   motor
motor   motor   motor   motor   motor
motor   motor   motor   motor   motor

*
```

０ＤＡＥＨ　内蔵リレーのコントロール

アドレス	０ＤＡＥＨ
機　　能	内蔵リレーのコントロール
レジスタ	ＡＦ－－－－－－－

解　　説　アキュムレータに格納されたデータが００Ｈの場合には内蔵リレーをＯＦＦ（ＭＯＴＯＲ　０）の状態に，アキュムレータに格納されたデータが００Ｈ以外の場合には内蔵リレーをＯＮ（ＭＯＴＯＲ　１）の状態に設定してもどります．

　ＰＣ－８００１ｍｋⅡでは，出力ポート３０Ｈ番地の第３ビットを直接に制御することで内蔵リレーをコントロールすることが可能ですが，この出力ポートにはリレーの他にもいくつかの周辺機器が接続されています．したがって，この出力ポートへの出力を実行する場合には，第３ビット以外のビットが変化しないようにＥＡ６６Ｈ番地に格納された値（以前３０Ｈ番地に出力したデータ）との間で，論理和（ＯＲ）を求めるなどの配慮が必要です．

サンプル

```
                ;
                ; --- control motor relay ---
                ;
                        ORG   0C100H
                ;
C100 2140C1     START:  LD    HL,PROMPT
C103 CD37C1             CALL  DSPMSG
C106 CD750F             CALL  0F75H           ; input a character from keyboard
C109 FE03               CP    03H             ; check stop key
C10B CA665C             JP    Z,5C66H         ; jump to system monitor if stop key
                ;
C10E FE30               CP    '0'
C110 200B               JR    NZ,NOTOFF
                ;
C112 2158C1             LD    HL,MSG0         ;
C115 CD37C1             CALL  DSPMSG          ; motor relay off
C118 CD2CC1             CALL  MOTOR0          ;
                ;
C11B 18E3               JR    START
                ;
C11D FE31       NOTOFF:CP     '1'
C11F 20DF               JR    NZ,START
                ;
C121 216FC1             LD    HL,MSG1         ;
C124 CD37C1             CALL  DSPMSG          ; motor relay on
C127 CD31C1             CALL  MOTOR1          ;
                ;
C12A 18D4               JR    START
                ;
                ; motor relay off subroutine
                ;
                ;    destorys : register a,f
                ;
```

```
C12C AF         MOTOR0:XOR  A
C12D CDAE0D            CALL 0DAEH              ;*control motor relay
C130 C9                RET
                ;
                ; motor relay on subroutine
                ;
                ;    destroys : register a,f
                ;
C131 3E01       MOTOR1:LD   A,1
C133 CDAE0D            CALL 0DAEH              ;*control motor relay
C136 C9                RET
                ;
                ; display message subroutine
                ;
                ;    inputs   : top address of message in register hl
                ;    destroys : register h,l
                ;
C137 7E         DSPMSG:LD   A,(HL)
C138 A7                AND  A
C139 C8                RET  Z
C13A CD5702            CALL 0257H              ; display a character
C13D 23                INC  HL
C13E 18F7              JR   DSPMSG
                ;
                ; message area
                ;
C140 0D0A6D6F   PROMPT:DEFB 0DH,0AH,'motor 0 or motor 1 ? ',0
C144 746F7220
C148 30206F72
C14C 206D6F74
C150 6F722031
C154 203F2000
C158 30203A20   MSG0:  DEFB '0 : motor relay off !!',0
C15C 6D6F746F
C160 72207265
C164 6C617920
C168 6F666620
C16C 212100
C16F 31203A20   MSG1:  DEFB '1 : motor relay on !!',0
C173 6D6F746F
C177 72207265
C17B 6C617920
C17F 6F6E2021
C183 2100
                ;
C185                   END


*GC100

motor 0 or motor 1 ?
motor 0 or motor 1 ?
motor 0 or motor 1 ?
motor 0 or motor 1 ?
motor 0 or motor 1 ?
motor 0 or motor 1 ?
motor 0 or motor 1 ?
motor 0 or motor 1 ? 0 : motor relay off !!
motor 0 or motor 1 ? 1 : motor relay on !!
motor 0 or motor 1 ? 0 : motor relay off !!
motor 0 or motor 1 ? 1 : motor relay on !!
motor 0 or motor 1 ? 0 : motor relay off !!
motor 0 or motor 1 ? 1 : motor relay on !!
motor 0 or motor 1 ? 0 : motor relay off !!
motor 0 or motor 1 ? 1 : motor relay on !!
motor 0 or motor 1 ? 0 : motor relay off !!
motor 0 or motor 1 ? 1 : motor relay on !!
motor 0 or motor 1 ? 0 : motor relay off !!
motor 0 or motor 1 ? 1 : motor relay on !!
motor 0 or motor 1 ?
motor 0 or motor 1 ?
motor 0 or motor 1 ?
*
```

0F75H　キーボードからの1バイト入力待ち

アドレス　0F75H

機　能　キーボードからの1バイト入力待ち

レジスタ　AF------

解　説　キーボードからの入力が認められるまで待ち，入力があった場合には，入力されたコードをアキュムレータに格納してもどります．ただし，CRTスクリーンへのエコー・バックは行いません．

キーボードからの入力がなければ入力があるまで待ち続けます．

サンプル

```
               ;
               ; --- input a character and display code ---
               ;
                      ORG   0C100H
               ;
C100 CD750F    LOOP:  CALL  0F75H          ;*input character from keyboard
C103 F5               PUSH  AF
               ;
C104 CD18C1           CALL  DSPACC         ; display character code
               ;
C107 3E20             LD    A,' '          ;
C109 CD5702           CALL  0257H          ; display two spaces
C10C CD5702           CALL  0257H          ;
               ;
C10F F1               POP   AF
C110 FE03             CP    03H            ; check break code
C112 CA665C           JP    Z,5C66H        ; jump to system monitor if stop key
               ;
C115 C300C1           JP    LOOP
               ;
               ; display accumulator hexa decimal subroutine
               ;
               ;    inputs   : binary code in accumulator
               ;    destroys : register a,f,b
               ;
C118 47        DSPACC:LD    B,A
C119 0F               RRCA
C11A 0F               RRCA
C11B 0F               RRCA
C11C 0F               RRCA
C11D CD21C1           CALL  HALF
C120 78               LD    A,B
               ;
C121 E60F      HALF:  AND   0FH
C123 FE0A             CP    10
C125 3802             JR    C,NUMBER
C127 C607             ADD   A,7
               ;
C129 C630      NUMBER:ADD   A,30H
C12B CD5702           CALL  0257H          ; display a character
C12E C9               RET
               ;
C12F                  END


*GC100
4F  50  51  52  53  54  55  56  57  58
59  5A  03
*
```

０Ｆ７ＢＨ　キーボードからの１バイト入力

アドレス　０Ｆ７ＢＨ

機　能　キーボードからの１バイト入力

レジスタ　ＡＦ－－－－－－

解　説　キーボードからの入力を調べ，キー入力が認められない場合には，キャリ・フラグ（ＣＹ）を１にセットしてもどります．キー入力が認められた場合にはキャリ・フラグ（ＣＹ）を０にリセットし，入力したコードをアキュムレータに格納してもどります．

０Ｆ７５Ｈ番地のシステム・サブルーチンではキー入力が認められるまで待ち続けますが，このシステム・サブルーチンではキー入力の有無にかかわらずもどってくるため，見掛け上の同時処理を行うことができます．

ただし，特殊キー（ＣＴＲＬキー，ＳＨＩＦＴキー，カナキー，ＧＲＰＨキー等）の入力がある場合には，処理速度が著しく低下します．リアルタイムな処理中で使用する場合には十分に御注意下さい．

サンプル

```
                 ;
                 ; --- key scanning with display cursor ---
                 ;
                         ORG  0C100H
                 ;
C100 2138C1      LOOPL:  LD   HL,MESAGE
C103 CD2FC1              CALL DSPMSG
                 ;
C106 0614                LD   B,20              ; set loop counter
C108 CD7B0F      LOOPS:  CALL 0F7BH             ;*key scanning with display cursor
C10B 300C                JR   NC,YESIN
                 ;
C10D CD26C1              CALL WAIT              ; wait a moment
                 ;
C110 3E3E                LD   A,'>'
C112 CD5702              CALL 0257H             ; display a character
                 ;
C115 10F1                DJNZ LOOPS
                 ;
C117 18E7                JR   LOOPL
                 ;
C119 F5          YESIN:  PUSH AF
C11A 3E20                LD   A,' '
C11C CD5702              CALL 0257H
C11F F1                  POP  AF
C120 CD5702              CALL 0257H             ; echo-back a input character
C123 C3665C              JP   5C66H             ; jump to system monitor
                 ;
                 ; wait subroutine
                 ;
                 ;    destroys : register a,f,h,l
                 ;
C126 21FF0F      WAIT:   LD   HL,0FFFH
```

```
C129 2B        WAIT1: DEC  HL
C12A 7C               LD   A,H
C12B B5               OR   L
C12C 20FB             JR   NZ,WAIT1
C12E C9               RET
               ;
               ; display message subroutine
               ;
               ;    inputs   : top address of message in register hl
               ;    destroys : register a,f,h,l
               ;
C12F 7E        DSPMSG:LD   A,(HL)
C130 A7               AND  A
C131 C8               RET  Z
C132 CD5702           CALL 0257H              ; display a character
C135 23               INC  HL
C136 18F7             JR   DSPMSG
               ;
               ; message area
               ;
C138 0D0A706C  MESAGE:DEFB 0DH,0AH,'please hit any key ',0
C13C 65617365
C140 20686974
C144 20616E79
C148 206B6579
C14C 2000
               ;
C14E                  END


*GC100

please hit any key >>>>>>>>>>>>>>>>>>>>
please hit any key >>>>>>>>>>>>>>>>>>>>
please hit any key >>>>>>>>>>>>>>>>>>>>
please hit any key >>>>>>>>>>>>>>>>>>>>
please hit any key >>>>>>>>>>>>>>>>>>>>
please hit any key >>>>>>>>>>>>>>>>>>>>
please hit any key >>>>>>>>>>>>>>>>>>>>
please hit any key >>>>>>>>>>>>>>>>>>>>
please hit any key >>>>>>>>>>>>>>>>>>>>
please hit any key >>>>>>>>>>>>>>>>>>>>
please hit any key >>>>>>>>>>>>>>>>>>>>
please hit any key >>>>>>>>>>>>>>>>>>>>
please hit any key >>>>>>>>>>>>>>>>>>>>
please hit any key >>>>>>>>>>>>>>>>>>>>
please hit any key >>>>>>>>>>>>>>>>>>>>
please hit any key >>>>>>>>>>>>>>>>>>>>
please hit any key >>>>>>>>>>>>>>>>>>>>
please hit any key >>>>>>>>>>>>>>>>>>>>
please hit any key >>>>>>>>>>>>>>>>>>>>
please hit any key >>>>>>>>>>>>>> x
*
```

０ＦＡＣＨ リアルタイム・キーボード・スキャニング

アドレス	０ＦＡＣＨ
機　能	リアルタイム・キーボード・スキャニング
レジスタ	ＡＦ－－－－－－

解　説 カーソルを表示せずにキーボードからの入力を調べ，キー入力が認められない場合にキャリ・フラグ（ＣＹ）を１にセットしてもどります．キー入力が認められた場合にはキャリ・フラグ（ＣＹ）を０にリセットし，入力したコードをアキュムレータに格納してもどります．

基本的には，ＢＡＳＩＣのＩＮＫＥＹ＄と同様の使い方を実施するための，サブルーチンです．

サンプル

```
               ;
               ; --- key scanning without display cursor ---
               ;
                       ORG   0C100H
               ;
C100 2138C1    LOOPL:  LD    HL,MESAGE
C103 CD2FC1            CALL  DSPMSG
               ;
C106 0614              LD    B,20              ; set loop counter
C108 CDAC0F    LOOPS:  CALL  0FACH             ;*key scanning without cursor
C10B 300C              JR    NC,YESIN          ;
               ;
C10D CD26C1            CALL  WAIT              ; wait a moment
               ;
C110 3E3E              LD    A,'>'
C112 CD5702            CALL  0257H             ; display a character
               ;
C115 10F1              DJNZ  LOOPS
               ;
C117 18E7              JR    LOOPL
               ;
C119 F5        YESIN:  PUSH  AF
C11A 3E20              LD    A,' '
C11C CD5702            CALL  0257H
C11F F1                POP   AF
C120 CD5702            CALL  0257H             ; echo-back a input character
C123 C3665C            JP    5C66H             ; jump to system monitor
               ;
               ; wait subroutine
               ;
               ;    destroys : register a,f,h,l
               ;
C126 21FF0F    WAIT:   LD    HL,0FFFH
C129 2B        WAIT1:  DEC   HL
C12A 7C                LD    A,H
C12B B5                OR    L
C12C 20FB              JR    NZ,WAIT1
C12E C9                RET
               ;
               ; display message subroutine
               ;
               ;    inputs   : top address of message in register hl
               ;    destroys : register a,f,h,l
               ;
```

```
C12F 7E        DSPMSG:LD   A,(HL)
C130 A7               AND  A
C131 C8               RET  Z
C132 CD5702           CALL 0257H          ; display a character
C135 23               INC  HL
C136 18F7             JR   DSPMSG
               ;
               ; message area
               ;
C138 0D0A706C MESAGE:DEFB 0DH,0AH,'please hit any key ',0
C13C 65617365
C140 20686974
C144 20616E79
C148 206B6579
C14C 2000
               ;
C14E                  END


*GC100

please hit any key >>>>>>>>>>>>>>>>>>>>
please hit any key >>>>>>>>>>>>>>>>>>>>
please hit any key >>>>>>>>>>>>>>>>>>>>
please hit any key >>>>>>>>>>>>>>>>>>>>
please hit any key >>>>>>>>>>>>>>>>>>>>
please hit any key >>>>>>>>>>>>>>>>>>>>
please hit any key >>>>>>>>>>>>>>>>>>>>
please hit any key >>>>>>>>>>>>>>>>>>>>
please hit any key >>>>>>>>>>>>>>>>>>>>
please hit any key >>>>>>>>>>>>>>>>>>>>
please hit any key >>>>>>>>>>>>>>>>>>>>
please hit any key >>>>>>>>>>>>>>>>>>>>
please hit any key >>>>>>>>>>>>>>>>>>>>
please hit any key >>>>>>>>>>>>>>>>>>>>
please hit any key >>>>>>>>>>>>>>>>>>>>
please hit any key >>>>>>>>>>>>>>>>>>>>
please hit any key >>>>>>>>>>>>>>>>>>>>
please hit any key >>>>>>>>>>>>>>>>>>>>
please hit any key >>>>>>>>>>>>>>>>>>>>
please hit any key >>>>>>>>>>>>>>>>>>>>
please hit any key >>>>>>>>>>>>>>>>>>>>
please hit any key >>>>> +
*
```

１２４ＡＨ　テキスト・スクリーン・コピー

アドレス	１２４ＡＨ
機　能	テキスト・スクリーン・コピー
レジスタ	ＡＦＢＣＤＥＨＬ

解　説　ＣＲＴのテキスト画面に表示されているキャラクタのイメージ（テキスト画面の最上段からスクロール範囲の下端まで）をそのままプリンタに出力します.

ただし，ヌル・キャラクタ・コードは２０Ｈである必要があり，アトリビュート指定は無視されるため，セミ・グラフィックによる描画画面は正常にコピーできません.

また，ＥＡ５８Ｈ番地の最下位ビット（ＬＳＢ）をセットしておき，［Ｆ・９］キーを押下した場合にも，同様にテキスト・スクリーンのコピーを行うことができます.

サンプル

```
                ;
                ; --- print out y bar graph ---
                ;
                        ORG   0C100H
                ;
C100 CD5A04     BARGRP:CALL  045AH          ; clear text screen all
                ;
C103 0E24               LD    C,36
C105 2601               LD    H,1
                ;
C107 0610               LD    B,16
C109 2E13               LD    L,19
                ;
C10B 3E21               LD    A,'!'
C10D CD50C1     LOOP:   CALL  MOVDSP
C110 2D                 DEC   L
C111 10FA               DJNZ  LOOP
                ;
C113 24                 INC   H             ; increment position x
C114 0D                 DEC   C
                ;
C115 CD88C1     LOOPX:  CALL  RND
C118 2E13               LD    L,19
                ;
C11A 3E2D               LD    A,'-'
C11C CD50C1             CALL  MOVDSP        ; move cursor and display character
C11F 2D                 DEC   L             ; decrement position y
                ;
C120 04                 INC   B
C121 05         LOOPY:  DEC   B
C122 2808               JR    Z,EXITY
                ;
C124 3E2A               LD    A,'*'
C126 CD50C1             CALL  MOVDSP        ; move cursor and display character
C129 2D                 DEC   L             ; decrement position y
C12A 18F5               JR    LOOPY
                ;
```

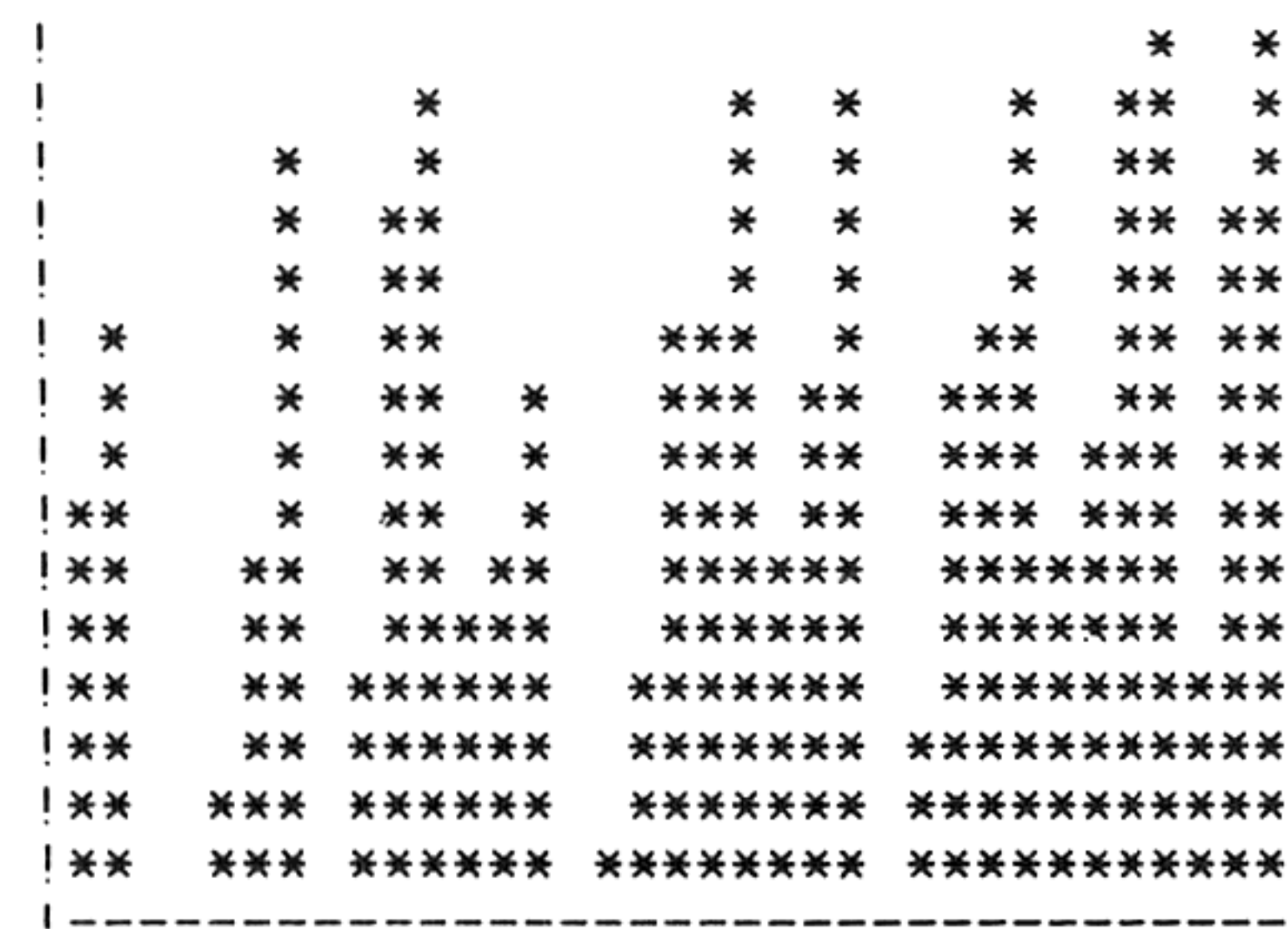
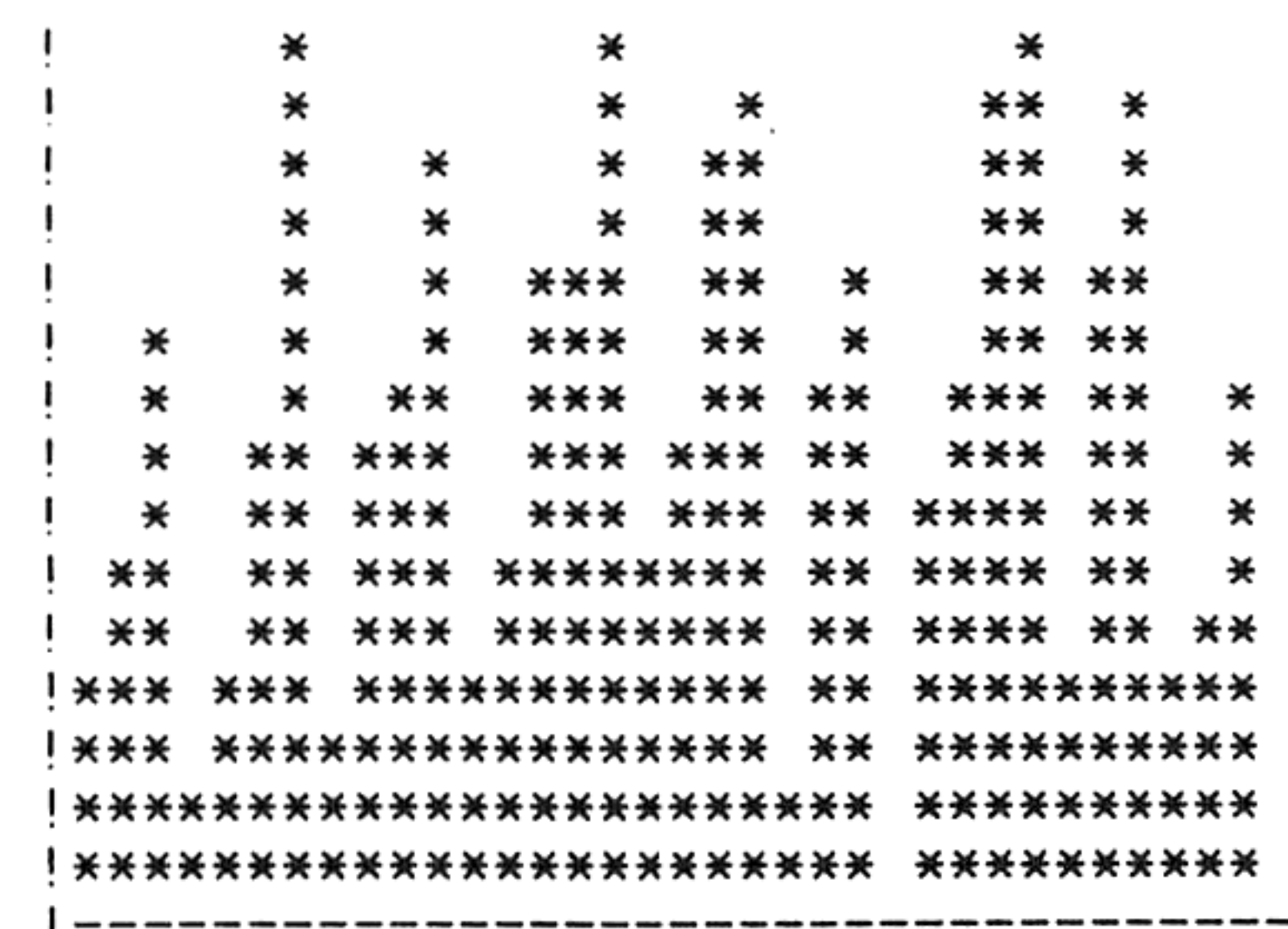
```
C12C 24         EXITY: INC  H                   ; increment position x
C12D 0D                DEC  C
C12E 20E5              JR   NZ,LOOPX
                ;
C130 2163C1     LOOPIN:LD   HL,MESAGE
C133 CD5AC1            CALL DSPMSG
                ;
C136 CD750F            CALL 0F75H               ; input a character from keyboard
C139 CD5702            CALL 0257H               ; echo-back a character
C13C CBEF              SET  5,A                 ; convert into small letter
                ;
C13E FE65              CP   'e'
C140 CA665C            JP   Z,5C66H
C143 FE6E              CP   'n'
C145 28B9              JR   Z,BARGRP
C147 FE70              CP   'p'
C149 20E5              JR   NZ,LOOPIN
                ;
                ; print out y bar graph section
                ;
C14B CD4A12            CALL 124AH               ;*print out text screen image
C14E 18E0              JR   LOOPIN
                ;
                ; move cursor and display a character subroutine
                ;
                ;    inputs   : position x in register h
                ;               position y in register l
                ;               character code in accumulator
                ;    destroys : register d,e
                ;
C150 F5         MOVDSP:PUSH AF
C151 C5                PUSH BC
C152 E5                PUSH HL
C153 CDD702            CALL 02D7H
C156 E1                POP  HL
C157 C1                POP  BC
C158 F1                POP  AF
C159 C9                RET
                ;
                ; display message subroutine
                ;
                ;    inputs   : top address of message in register hl
                ;    destroys : register a,f,h,l
                ;
C15A 7E         DSPMSG:LD   A,(HL)
C15B A7                AND  A
C15C C8                RET  Z
C15D CD5702            CALL 0257H               ; display a character
C160 23                INC  HL
C161 18F7              JR   DSPMSG
                ;
                ; message area
                ;
C163 0B         MESAGE:DEFB 11
C164 6E657874          DEFB 'next graph(n) '
C168 20677261
C16C 7068286E
C170 2920
C172 7072696E          DEFB 'print(p) '
C176 74287029
C17A 20
C17B 6F722065          DEFB 'or end(e) ? ',0
C17F 6E642865
C183 29203F20
C187 00
                ;
                ; generate a random number from 0 to 15 subroutine
                ;
                ;    outputs  : generated number in register b
                ;    destroys : register a,f,b
                ;
C188 CD8FC1     RND:   CALL RSUB
C18B E60F              AND  0FH
C18D 47                LD   B,A
```

```
C18E C9               RET
                ;
C18F D5         RSUB: PUSH DE
C190 E5               PUSH HL
C191 ED5F             LD   A,R
C193 1600             LD   D,0
C195 5F               LD   E,A
C196 2AA9C1           LD   HL,(RNDM)
C199 19               ADD  HL,DE
C19A 22A9C1           LD   (RNDM),HL
C19D 7D               LD   A,L
C19E CB0F             RRC  A
C1A0 CB0F             RRC  A
C1A2 CB0F             RRC  A
C1A4 CB0F             RRC  A
C1A6 E1               POP  HL
C1A7 D1               POP  DE
C1A8 C9               RET
                ;
C1A9            RNDM: END
```

```
*GC100
next graph(n) print(p) or end(e) ? p


!                               *  *
!          *        *  *    *  **  *
!      *   *        *  *    *  **  *
!      *  **        *  *    *  ** **
!      *  **        *  *    *  ** **
! *    *  **      ***  *   **  ** **
! *    *  **  *   *** **  ***  ** **
! *    *  **  *   *** **  *** *** **
!**    *  **  *   *** **  *** *** **
!**   **  ** **   ******  ******* **
!**   **  *****   ******  ******* **
!**   ** ******  *******  **********
!**   ** ******  ******* ***********
!**  *** ******  ******* ***********
!**  *** ****** ******** ***********
!-----------------------------------


next graph(n) print(p) or end(e) ? p


!      *        *           *
!      *        *   *      **  *
!      *   *    *  **      **  *
!      *   *    *  **      **  *
!      *   *  ***  **  *   ** **
!  *   *   *  ***  **  *   ** **
!  *   *  **  ***  ** **  *** **  *
!  *  ** ***  *** *** **  *** **  *
!  *  ** ***  *** *** ** **** **  *
! **  ** *** ******** ** **** **  *
! **  ** *** ******** ** **** ** **
!*** *** ************ ** **********
!*** **************** ** **********
!*********************** **********
!*********************** **********
!-----------------------------------
```

１６０２Ｈ　タイマからのデータ読出し

アドレス	１６０２Ｈ
機　　能	タイマからのデータ読出し
レジスタ	ＡＦＢＣＤＥＨＬ

解　　説　μＰＤ１９９０Ｃ（シリアルＩ／Ｏカレンダ時計）からの情報を読出し，ワーク・エリア（ＥＡ７６Ｈ～ＥＡ７ＢＨ番地）に格納してもどります．

それぞれのデータはＢＣＤ（Binary Coded Decimal）コードとしてワーク・エリアに格納されますが，ＭＯＮＴＨ（ＥＡ７ＡＨ番地）のみは例外で，通常のバイナリ形式（０１Ｈ～０ＣＨ）で格納されます．

以下に，タイマ制御用の情報読出しバッファを示します．

ＥＡ７６Ｈ番地……ＳＥＣＯＮＤ（ＢＣＤコードによる秒）
ＥＡ７７Ｈ番地……ＭＩＮＵＴＥ（ＢＣＤコードによる分）
ＥＡ７８Ｈ番地……ＨＯＵＲ　　（ＢＣＤコードによる時）
ＥＡ７９Ｈ番地……ＤＡＹ　　　（ＢＣＤコードによる日）
ＥＡ７ＡＨ番地……ＭＯＮＴＨ　（バイナリ・コードによる月）
ＥＡ７ＢＨ番地……ＹＥＡＲ　　（ＢＣＤコードによる年）

ただし，データそれぞれの有効範囲はつぎのとおりです．

ＳＥＣＯＮＤ……０～５９
ＭＩＮＵＴＥ……０～５９
ＨＯＵＲ…………０～２３
ＤＡＹ……………１～３１
ＭＯＮＴＨ………１～１２
ＹＥＡＲ…………０～９９

ＹＥＡＲ（年）の値は，タイマからの読出しを行っていません．したがって，インタプリタは，ＹＥＡＲ（年）の値を参照する必要ができた場合にはタイマからのデータ読出しを行わず，ＥＡ７ＢＨ番地の内容を参照しています．

サンプル

```
                ;
                ; --- read from timer ---
                ;
                        ORG  0C100H
                ;
                ; read from timer section
                ;
C100 CD0216             CALL 1602H          ;*read from timer
```

```
                        ;
                        ; display date section
                        ;
C103 2175C1                     LD   HL,ADATE
C106 CD6CC1                     CALL DSPMSG
                        ;
C109 217BEA                     LD   HL,0EA7BH        ; end address of date buffer
                        ;
C10C 0603                       LD   B,3
C10E 7E                 LDATE:  LD   A,(HL)
C10F 2B                         DEC  HL
                        ;
C110 48                         LD   C,B
C111 0D                         DEC  C
C112 0D                         DEC  C
C113 2805                       JR   Z,MONTH          ; check minute or other
                        ;
C115 CD3BC1                     CALL BCDCHR           ; convert bcd code into 2 characters
C118 1803                       JR   SKIP
                        ;
C11A CD4EC1             MONTH:  CALL CMONTH           ; convert month into 2 characters
                        ;
C11D CD5EC1             SKIP:   CALL DSPDE            ; display two characters in reg-de
                        ;
C120 10EC                       DJNZ LDATE
                        ;
                        ; display time section
                        ;
C122 218CC1                     LD   HL,ATIME
C125 CD6CC1                     CALL DSPMSG
                        ;
C128 2178EA                     LD   HL,0EA7BH-3      ; end address of time buffer
                        ;
C12B 0603                       LD   B,3
C12D 7E                 LTIME:  LD   A,(HL)
C12E 2B                         DEC  HL
                        ;
C12F CD3BC1                     CALL BCDCHR           ; convert bcd into 2 characters
C132 CD5EC1                     CALL DSPDE            ; display two characters in reg-de
C135 1B                         DEC  DE
C136 10F5                       DJNZ LTIME
                        ;
C138 C3665C                     JP   5C66H            ; jump to system monitor
                        ;
                        ; convert bcd code into character code subroutine
                        ;
                        ;    inputs   : bcd code in accumulator
                        ;    outputs  : two characters in register de
                        ;    destroys : register a,f,d,e
                        ;
C13B 5F                 BCDCHR:LD   E,A
C13C CB3F                       SRL  A
C13E CB3F                       SRL  A
C140 CB3F                       SRL  A
C142 CB3F                       SRL  A
C144 C630                       ADD  A.'0'
C146 57                         LD   D,A
                        ;
C147 7B                         LD   A,E
C148 E60F                       AND  0FH
C14A C630                       ADD  A,'0'
C14C 5F                         LD   E,A
C14D C9                         RET
                        ;
                        ; convert month into character code subroutine
                        ;
                        ;    inputs   : month in accumulator
                        ;    outputs  : two characters in register de
                        ;    destroys : register a,f,d,e
                        ;
C14E 1630               CMONTH:LD   D,'0'
C150 FE0A                       CP   0AH
C152 3004                       JR   NC,CMONT1
C154 C630                       ADD  A,'0'
```

```
C156 5F               LD   E,A
C157 C9               RET
              ;
C158 1631     CMONT1:LD   D,'1'
C15A C626             ADD  A,'0'-0AH
C15C 5F               LD   E,A
C15D C9               RET
              ;
              ; display two characters and a space
              ;
              ;    destroys : accumulator only
              ;
C15E 7A       DSPDE: LD   A,D
C15F CD5702          CALL 0257H          ; display a character
C162 7B              LD   A,E
C163 CD5702          CALL 0257H          ; display a charecter
C166 3E20            LD   A,' '
C168 CD5702          CALL 0257H          ; display a space code
C16B C9              RET
              ;
              ; display message subroutine
              ;
              ;    destroys  : register a,f,h,l
              ;
C16C 7E       DSPMSG:LD   A,(HL)
C16D A7              AND  A
C16E C8              RET  Z
C16F CD5702          CALL 0257H          ; display a character
C172 23              INC  HL
C173 18F7            JR   DSPMSG
              ;
              ; message area
              ;
C175 0D0A6461 ADATE: DEFB 0DH,0AH,'date - yy mm dd --> ',0
C179 7465202D
C17D 20797920
C181 6D6D2064
C185 64202D2D
C189 3E2000
C18C 0D0A7469 ATIME: DEFB 0DH,0AH,'time - hh mm ss --> ',0
C190 6D65202D
C194 20686820
C198 6D6D2073
C19C 73202D2D
C1A0 3E2000
              ;
C1A3                 END
```

```
print date$,time$
79/01/01      00:00:56
Ok
mon
*GC100

date - yy mm dd --> 79 01 01
time - hh mm ss --> 00 01 00
*
Ok
date$="01/01/01":time$="01:01:01"
Ok
mon
*GC100

date - yy mm dd --> 01 01 01
time - hh mm ss --> 01 01 11
*
Ok
print date$,time$
01/01/01      01:01:23
Ok
```

１６６３H　タイマへのデータ書込み

アドレス	１６６３H
機　能	タイマへのデータ書込み
レジスタ	AFBCDEHL

解　説　ワーク・エリア（EA７６H～EA７BH番地）の情報を，μPD１９９０C（シリアルI／Oカレンダ時計）に設定してもどります．

それぞれのデータは，BCD（Binary Coded Decimal）コードとしてワーク・エリアに与える必要がありますが，MONTH（EA７AH番地）のみは例外で，通常のバイナリ形式（０１H～０CH）で与えます．

以下に，タイマ制御用の情報書込みバッファを示します．

EA７６H番地……SECOND（BCDコードによる秒）
EA７７H番地……MINUTE（BCDコードによる分）
EA７８H番地……HOUR（BCDコードによる時）
EA７９H番地……DAY（BCDコードによる日）
EA７AH番地……MONTH（バイナリ・コードによる月）
EA７BH番地……YEAR（BCDコードによる年）

ただし，データそれぞれの有効範囲はつぎのとおりです．

SECOND……０～５９
MINUTE……０～５９
HOUR…………０～２３
DAY……………１～３１
MONTH………１～１２
YEAR…………０～９９

YEAR（年）の値は，タイマへの書込みを行っていません．したがって，インタプリタは，YEAR（年）の値を参照する必要ができた場合にはタイマからのデータ読出しを行わず，EA７BH番地の内容を参照しています．

サンプル

```
             ;
             ; --- write to timer ---
             ;
                   ORG  0C100H
             ;
             ; input date section
             ;
C100 2175C1        LD   HL,PDATE
```

```
C103 CD6CC1          CALL DSPMSG
C106 CD7E1B          CALL 1B7EH           ; screen editor
C109 DA665C          JP   C,5C66H         ; jump to system monitor if stop key
                ;
C10C 117BEA          LD   DE,0EA7BH       ; end address of date buffer
                ;
C10F 0603            LD   B,3
C111 C5         LDATE: PUSH BC
                ;
C112 23              INC  HL
C113 46              LD   B,(HL)
C114 23              INC  HL
C115 4E              LD   C,(HL)
                ;
C116 F1              POP  AF              ;
C117 F5              PUSH AF              ;
C118 3D              DEC  A               ; check minute or other
C119 3D              DEC  A               ;
C11A 2805            JR   Z,MONTH         ;
                ;
C11C CD4DC1          CALL CHRBCD          ; convert into bcd code if not month
C11F 1803            JR   SKIP
                ;
C121 CD5EC1     MONTH: CALL CHRBIN        ; convert into binary code if month
                ;
C124 12         SKIP:  LD   (DE),A
C125 1B              DEC  DE
                ;
C126 C1              POP  BC
C127 10E8            DJNZ LDATE
                ;
                ; input time section
                ;
C129 2186C1          LD   HL,PTIME
C12C CD6CC1          CALL DSPMSG
C12F CD7E1B          CALL 1B7EH           ; screen editor
C132 DA665C          JP   C,5C66H         ; jump to system monitor if stop key
                ;
C135 1178EA          LD   DE,0EA7BH-3     ; end address of date buffer
                ;
C138 0603            LD   B,3
C13A C5         LTIME: PUSH BC
                ;
C13B 23              INC  HL
C13C 46              LD   B,(HL)
C13D 23              INC  HL
C13E 4E              LD   C,(HL)
                ;
C13F CD4DC1          CALL CHRBCD          ; convert into bcd code
                ;
C142 12              LD   (DE),A
C143 1B              DEC  DE
                ;
C144 C1              POP  BC
C145 10F3            DJNZ LTIME
                ;
                ; write to timer section
                ;
C147 CD6316          CALL 1663H           ;*write to timer
                ;
C14A C3665C          JP   5C66H           ; jump to system monitor
                ;
                ; convert character code into bcd code subroutine
                ;
                ;    inputs   : two characters in register bc
                ;    outputs  : bcd code in accumulator
                ;    destroys : register a,f,b
                ;
C14D 78         CHRBCD:LD   A,B
C14E D630            SUB  '0'
C150 47              LD   B,A
C151 CB20            SLA  B
C153 CB20            SLA  B
C155 CB20            SLA  B
```

```
C157 CB20               SLA  B
C159 79                 LD   A,C
C15A D630               SUB  '0'
C15C 80                 ADD  A,B
C15D C9                 RET
                ;
                ; convert character code into binary code subroutine
                ;
                ;    inputs   : two characters in register bc
                ;    outputs  : binary code in accumulator
                ;    destroys : register a,f,b
                ;
C15E 78         CHRBIN:LD   A,B
C15F D630               SUB  '0'
                ;
C161 47                 LD   B,A              ;
C162 87                 ADD  A,A              ;
C163 87                 ADD  A,A              ; multiply accumulator by ten
C164 80                 ADD  A,B              ;
C165 87                 ADD  A,A              ;
                ;
C166 47                 LD   B,A
C167 79                 LD   A,C
C168 D630               SUB  '0'
C16A 80                 ADD  A,B
C16B C9                 RET
                ;
                ; display message subroutine
                ;
                ;    destroys    : register a,f,h,l
                ;
C16C 7E         DSPMSG:LD   A,(HL)
C16D A7                 AND  A
C16E C8                 RET  Z
C16F CD5702             CALL 0257H            ; display a character
C172 23                 INC  HL
C173 18F7               JR   DSPMSG
                ;
                ; message area
                ;
C175 64617465 PDATE: DEFB 'date - yymmdd ? ',0
C179 202D2079
C17D 796D6D64
C181 64203F20
C185 00
C186 74696D65 PTIME: DEFB 'time - hhmmss ? ',0
C18A 202D2068
C18E 686D6D73
C192 73203F20
C196 00
                ;
C197                    END
```

```
print date$,time$
79/01/01        00:00:06
Ok
mon
*GC100
date - yymmdd ? 011223
time - hhmmss ? 122334

*
Ok
print date$,time$
01/12/23        12:23:42
Ok
mon
*GC100
date - yymmdd ? 831001
time - hhmmss ? 000000

 *
```

１７Ｅ９Ｈ　ＢＡＳＩＣのプログラム格納アドレス設定

アドレス	１７Ｅ９Ｈ
機　能	ＢＡＳＩＣのプログラム格納アドレス設定
レジスタ	ＡＦＢＣＤＥＨＬ

解　説　レジスタＨＬによって指定するアドレスを，ＢＡＳＩＣのプログラムを格納する先頭アドレス（ＲＡＭの下限）として設定し，ＢＡＳＩＣのタイトル・メッセージを表示してコールド・スタートを行います．

ただし，レジスタＨＬに与えるアドレスは，Ｚ８０のインタラプト・ベクタ・テーブルが置けるアドレス（２５６の倍数）の必要があります．

設定以後，インタプリタのスクリーン・エディタによって入力される，Ｎ－ＢＡＳＩＣおよびＮ８０－ＢＡＳＩＣのプログラムは，設定を行ったアドレス以降に格納されて行きます．

サンプル

```
                ;
                ; --- appoint top address of basic text ---
                ;
                        ORG   0C100H
                ;
C100 1187C1             LD    DE,MESAGE
C103 CD7EC1             CALL  DSPMSG
C106 11A4C1             LD    DE,QUESTI
C109 CD7EC1             CALL  DSPMSG
                ;
C10C CD2AC1             CALL  INADDR            ; input top address of basic text
C10F DA665C             JP    C,5C66H           ; do nothing if break or error
                ;
C112 1187C1             LD    DE,MESAGE
C115 CD7EC1             CALL  DSPMSG
C118 11A7C1             LD    DE,ANSWER
C11B CD7EC1             CALL  DSPMSG
                ;
C11E CD56C1             CALL  DSPADR            ;
C121 11AAC1             LD    DE,CRLF           ; display address and crlf code
C124 CD7EC1             CALL  DSPMSG            ;
                ;
C127 C3E917             JP    17E9H             ;*appoint top address of basic text
                ;
                ; input address subroutine
                ;
                ;    outputs  : input address in register hl
                ;               set carry if exit with error or stop key
                ;               reset carry if exit with return key
                ;    destroys : register a,f,h,l
                ;
C12A 210000     INADDR:LD    HL,0
C12D CD75C1     INADD1:CALL  MONIN              ; input a character
                ;
C130 FE04               CP    04H               ; check stop key or not
C132 D8                 RET   C                 ; set carry if exit with stop key
                ;
C133 FE0D               CP    0DH               ; check return key or not
C135 C8                 RET   Z                 ; reset carry if exit with return ke
                ;
C136 FE30       CHKHEX:CP    '0'                ;
C138 D8                 RET   C                 ;
C139 FE47               CP    'F'+1             ;
C13B 3017               JR    NC,SETCY          ; set carry and return
C13D FE3A               CP    '9'+1             ;   if input not hexa code
C13F 3803               JR    C,YESHEX          ;
C141 FE41               CP    'A'               ;
C143 D8                 RET   C                 ;
                ;
C144 FE3A       YESHEX:CP    '9'+1              ;
C146 3802               JR    C,INADD2          ;
C148 C609               ADD   A,9               ;
C14A E60F       INADD2:AND   0FH                ;
C14C 29                 ADD   HL,HL             ; convert into binary code
C14D 29                 ADD   HL,HL             ;   and store in register hl
C14E 29                 ADD   HL,HL             ;
C14F 29                 ADD   HL,HL             ;
C150 85                 ADD   A,L               ;
C151 6F                 LD    L,A               ;
                ;
C152 18D9               JR    INADD1            ; next digit
                ;
C154 37         SETCY:  SCF
C155 C9                 RET
                ;
                ; display address subroutine
                ;
                ;    inputs   : address in register hl
                ;    destroys : register a,f
                ;
C156 7C         DSPADR:LD    A,H
C157 CD5BC1             CALL  ACCHEX
```

```
C15A 7D                 LD    A,L
                ;
C15B F5         ACCHEX:PUSH  AF
C15C 0F                RRCA
C15D 0F                RRCA
C15E 0F                RRCA
C15F 0F                RRCA
C160 CD68C1            CALL  HALF
C163 F1                POP   AF
C164 CD68C1            CALL  HALF
C167 C9                RET
                ;
C168 E60F       HALF:  AND   0FH
C16A FE0A              CP    10
C16C 3802              JR    C,NUMBER
C16E C607              ADD   A,7
                ;
C170 C630       NUMBER:ADD   A,'0'
C172 C35702            JP    0257H
                ;
                ; input a capital character subroutine
                ;
                ;    outputs  : character code in accumulator
                ;    destroys : register a,f
                ;
C175 CD750F     MONIN: CALL  0F75H           ; input a character from keyboard
C178 CDCC4C            CALL  4CCCH           ; convert into capital letter
C17B C35702            JP    0257H           ; echo-back a character and return
                ;
                ; display message subroutine
                ;
                ;    destroys : register a,f,d,e
                ;
C17E 1A         DSPMSG:LD    A,(DE)
C17F A7                AND   A
C180 C8                RET   Z
C181 CD5702            CALL  0257H           ; display a character
C184 13                INC   DE
C185 18F7              JR    DSPMSG
                ;
                ; message area
                ;
C187 0D0A746F   MESAGE:DEFB 0DH,0AH,'top address of basic text ',0
C18B 70206164
C18F 64726573
C193 73206F66
C197 20626173
C19B 69632074
C19F 65787420
C1A3 00
C1A4 3F2000     QUESTI:DEFB '? ',0
C1A7 3A2000     ANSWER:DEFB ': ',0
C1AA 0D0A00     CRLF:  DEFB 0DH,0AH,0
                ;
C1AD                   END


*GC100

top address of basic text ? 9000
top address of basic text : 9000
NEC PC-8001 BASIC Ver 1.3
Copyright 1979 (C) by Microsoft

Ok
```

1875H “Disk BASIC Feature” エラー

アドレス	1875H
機能	“Disk BASIC Feature”
レジスタ	AFBCDEHL

解説 実行の準備がなされていない状態でDisk－BASIC関係の命令を実行したことを示す，“Disk BASIC Feature”エラーを出力してインタプリタの制御下に入ります．

具体的には，エラー・コード２６をレジスタＥに格納後，３ＢＦ９Ｈ番地にジャンプしています．

インタプリタでは，Disk－BASICが起動していない時点でDisk－BASIC関係の命令が実行された場合に，ワーク・エリア中に置かれたジャンプ・テーブルを介して，この１８７５Ｈ番地にジャンプして，“Disk BASIC Feature”エラーを出力します．

サンプル

```
                ;
                ; --- output 'Disk BASIC Feature' error ---
                ;
                        ORG   0C100H
                ;
C100 2116C1     LOOP:   LD    HL,PROMPT
C103 CDED52             CALL  52EDH             ; display the message
                ;
C106 CD750F             CALL  0F75H             ; input a character from keyboard
C109 CD5702             CALL  0257H             ; echo-back the character
                ;
C10C CBEF               SET   5,A               ; convert into small letter
                ;
C10E FE79               CP    'y'
C110 C2665C             JP    NZ,5C66H          ; jump to system monitor if not yes
                ;
C113 C37518             JP    1875H             ;*output 'Disk BASIC Feature' error
                ;
                ; message area
                ;
C116 0D0A6F75   PROMPT:DEFB 0DH,0AH,'output error (y/n) ? ',0
C11A 74707574
C11E 20657272
C122 6F722028
C126 792F6E29
C12A 203F2000
                ;
C12E                    END
```

```
*GC100

output error (y/n) ? y
Disk BASIC Feature
Ok
```

1B7EH　スクリーン・エディタ

アドレス	1B7EH
機　能	スクリーン・エディタ
レジスタ	AFBCDEHL

解　説　キーボードから，スクリーン・エディット方式によって1ラインのデータを入力し，BASICのインプット・バッファ（EC96H～ED95H番地）に格納してもどります．ただし，入力データは254バイトまでが有効で，254バイトを超えて入力した場合には，先頭からの254バイトが入力データと見なされます．

データの入力は，RETURNキー（CTRL+M）またはSTOPキー（CTRL+C）の入力によって終了しますが，前者の場合にはキャリ・フラグ（CY）を0にリセットし，後者の入力によってスクリーン・エディットを中断した場合には，キャリ・フラグ（CY）を1にセットしてもどります．

このサブルーチンからのリターン時には，レジスタHLにEC95H（インプット・バッファの開始アドレス－1）が，アキュムレータには最後に入力されたキャラクタのキャラクタ・コード（03Hまたは0DH）が与えられます．

サンプル

```
                ;
                ; --- micro word processer ---
                ;
                        ORG   0C100H
                ;
C100 21C1C1             LD    HL,TITLE
C103 1806               JR    DSPLAY
                ;
C105 3E07       ERROR:  LD    A,07H
C107 DF                 RST   18H             ; output a character to each device
                ;
C108 21D8C1     START0:LD     HL,PROMPT
C10B CDBAC1     DSPLAY:CALL   DSPMSG
                ;
C10E CD7E1B     START:  CALL  1B7EH           ;*screen editor
C111 38FB               JR    C,START         ; screen edit again if stop key
                ;
C113 23                 INC   HL
C114 7E                 LD    A,(HL)          ; get the first data of edit
C115 CBAF               RES   5,A             ; convert into capital letter
C117 FE41               CP    'A'
C119 38EA               JR    C,ERROR
C11B FE5B               CP    'Z'+1
C11D 30E6               JR    NC,ERROR
                ;
C11F 21E2C1             LD    HL,TABLE        ; top address of command table
C122 0604               LD    B,4             ; set number of command
C124 1196EC     START1:LD     DE,0EC96H       ; input buffer of basic
C127 1A         START2:LD     A,(DE)
C128 CBAF               RES   5,A             ; convert into capital letter
C12A BE                 CP    (HL)
C12B 2007               JR    NZ,START3
```

```
C12D A7                 AND  A
C12E 2828               JR   Z,YESCOM         ; found command
C130 23                 INC  HL
C131 13                 INC  DE
C132 18F3               JR   START2
                ;
C134 34         START3:INC  (HL)
C135 35                 DEC  (HL)
C136 2803               JR   Z,START4
C138 23                 INC  HL
C139 18F9               JR   START3
                ;
C13B 23         START4:INC  HL
C13C 23                 INC  HL
C13D 23                 INC  HL
C13E 10E4               DJNZ START1
                ;
C140 1196EC     EDIT:   LD   DE,0EC96H        ; input buffer of basic
C143 1A                 LD   A,(DE)
C144 13                 INC  DE
C145 08                 EX   AF,AF'
C146 1A                 LD   A,(DE)
C147 13                 INC  DE
C148 FE3A               CP   ':'
C14A 20B9               JR   NZ,ERROR
C14C 08                 EX   AF,AF'
C14D CDB0C1             CALL CALADR           ; calculate top address of buffer
C150 EB                 EX   DE,HL
C151 01FF00             LD   BC,0FFH
C154 EDB0               LDIR
C156 18B6               JR   START
                ;
C158 23         YESCOM:INC  HL                ;
C159 5E                 LD   E,(HL)           ;
C15A 23                 INC  HL               ; get command address
C15B 56                 LD   D,(HL)           ;   and jump to each command
C15C EB                 EX   DE,HL            ;
C15D E9                 JP   (HL)             ;
                ;
                ; new command section
                ;
C15E 2100C2     NEW:    LD   HL,BUFFER
C161 AF                 XOR  A
C162 061A               LD   B,'z'-'a'+1
C164 77         NEW1:   LD   (HL),A
C165 24                 INC  H
C166 10FC               DJNZ NEW1
C168 189E               JR   START0
                ;
                ; print command section
                ;
C16A 3E01       PRINT:  LD   A,1
C16C 3249EB             LD   (0EB49H),A       ; select printer for output
                ;
C16F 3E61               LD   A,'a'            ; set first line number in acc.
C171 CDA1C1     PRINT1:CALL PRNLIN            ; print out a line to printer
                ;
C174 47                 LD   B,A
C175 CDF10C             CALL 0CF1H            ; check stop key
C178 388E               JR   C,START0         ; return to main routine  if break
C17A 78                 LD   A,B
                ;
C17B 3C                 INC  A                ; increment line number in acc.
C17C FE7B               CP   'z'+1            ; check line number end
C17E 20F1               JR   NZ,PRINT1
                ;
C180 AF                 XOR  A
C181 3249EB             LD   (0EB49H),A       ; select crt for output
C184 1882               JR   START0
                ;
                ; list command section
                ;
C186 3E61       LIST:   LD   A,'a'            ; set first line number
C188 CD9BC1     LIST1: CALL LSTLIN            ; display a line
```

```
                  ;
C18B 47                 LD   B,A
C18C CDF10C             CALL 0CF1H              ; check stop key
C18F DA08C1             JP   C,START0           ; stop key, return to main routine
C192 78                 LD   A,B
                  ;
C193 3C                 INC  A                  ; increment line number in acc.
C194 FE7B               CP   'z'+1              ; check line number end
C196 20F0               JR   NZ,LIST1
C198 C308C1             JP   START0
                  ;
                  ; list one line with line number subroutine
                  ;
                  ;    inputs   : line number from a to z in accumulator
                  ;    destroys : register h,l
                  ;
C19B DF           LSTLIN:RST  18H               ; output line number
C19C F5                 PUSH AF
C19D 3E3A               LD   A,':'
C19F DF                 RST  18H
C1A0 F1                 POP  AF
                  ;
                  ; list one line without line number subroutine
                  ;
                  ;    inputs   : line number from a to z in accumulator
                  ;    destroys : register h,l
                  ;
C1A1 F5           PRNLIN:PUSH AF
C1A2 CDB0C1             CALL CALADR             ; calculate top address of buffer
C1A5 CDBAC1             CALL DSPMSG
                  ;
C1A8 21DFC1             LD   HL,CRLF
C1AB CDBAC1             CALL DSPMSG
                  ;
C1AE F1                 POP  AF
C1AF C9                 RET
                  ;
                  ; calculate top address of buffer subroutine
                  ;
                  ;    inputs   : line number from a to z in accumulator
                  ;    outputs  : top address of buffer in register hl
                  ;    destroys : register a,f,h,l
                  ;
C1B0 CBAF         CALADR:RES  5,A
C1B2 D641               SUB  'A'
C1B4 2100C2             LD   HL,BUFFER
C1B7 84                 ADD  A,H
C1B8 67                 LD   H,A
C1B9 C9                 RET
                  ;
                  ; display message subroutine
                  ;
                  ;    inputs   : top address of message in register hl
                  ;    destroys : register a,f,h,l
                  ;
C1BA 7E           DSPMSG:LD   A,(HL)
C1BB A7                 AND  A
C1BC C8                 RET  Z
C1BD DF                 RST  18H                ; output a character
C1BE 23                 INC  HL
C1BF 18F9               JR   DSPMSG
                  ;
                  ; message area
                  ;
C1C1 776F7264 TITLE: DEFB 'word processing system
C1C5 2070726F
C1C9 63657373
C1CD 696E6720
C1D1 73797374
C1D5 656D20
C1D8 68657265 PROMPT:DEFB 'here..',0FCH
C1DC 2E2EFC
C1DF 0D0A00   CRLF:  DEFB 0DH,0AH,0
                  ;
```

```
                ; command table
                ;
C1E2 4C495354 TABLE: DEFB 'LIST',0
C1E6 00
C1E7 86C1            DEFW LIST
C1E9 5052494E        DEFB 'PRINT',0
C1ED 5400
C1EF 6AC1            DEFW PRINT
C1F1 4E455700        DEFB 'NEW',0
C1F5 5EC1            DEFW NEW
C1F7 53595354        DEFB 'SYSTEM',0
C1FB 454D00
C1FE 665C            DEFW 5C66H
                ;
                ; buffer for edit
                ;
C200            BUFFER:END
```

```
<<< MICRO WORD PROCESSER SPECIFICATIONS >>>

 << BASIC SPECIFICATIONS >>

    program area            :       from c100h to c1ffh
    data area               :       from c200h to dbffh

    operation               :       screen edit
    line number             :       from a to z, or from A to Z

    title message           :       micro processing system here..
    prompt message          :       here..
    error message           :       ring the bell

 << COMMANDS >>

    list, or LIST           :       display all text
    print, or PRINT         :       output all text to printer
    new, or NEW             :       clear all text
    system, or SYSTEM       :       exit from word processing system

 << ATTENTION >>

    this specifications printed with this program.
```

1B8AH　ライン・エディタ

アドレス	1B8AH
機　能	ライン・エディタ
レジスタ	AFBCDEHL

解　説　キーボードから，ライン・エディト方式によって１ラインのデータを入力し，ＢＡＳＩＣのインプット・バッファ（ＥＣ９６Ｈ～ＥＤ９５Ｈ番地）に格納してもどります．ただし，入力データは２５４バイトまでが有効で，２５４バイトを超えて入力した場合には，先頭からの２５４バイトが入力データと見なされます．

データの入力は，ＲＥＴＵＲＮキー（ＣＴＲＬ＋Ｍ）またはＳＴＯＰキー（ＣＴＲＬ＋Ｃ）の入力によって終了しますが，前者の場合にはキャリ・フラグ（ＣＹ）を０にリセットし，後者の入力によってスクリーン・エディトを中断した場合には，キャリ・フラグ（ＣＹ）を１にセットしてもどります．

このサブルーチンからのリターン時には，レジスタＨＬにＥＣ９５Ｈ（インプット・バッファの開始アドレス－１）が，アキュムレータには最後に入力されたキャラクタのキャラクタ・コード（０３Ｈまたは０ＤＨ）が与えられます．

サンプル

```
                ;
                ; --- convert string into capitals ---
                ;
                        ORG   0C100H
                ;
C100 112FC1     START:  LD    DE,PROMPT
C103 CD26C1             CALL  DSPMSG
                ;
C106 CD8A1B             CALL  1B8AH           ; line input subroutine
C109 DA665C             JP    C,5C66H         ; jump to system monitor if stop key
                ;
C10C 113DC1             LD    DE,ANSWER
C10F CD26C1             CALL  DSPMSG
                ;
C112 23         LOOP:   INC   HL
C113 7E                 LD    A,(HL)          ; get the first data of edit
C114 A7                 AND   A               ; check end mark or not
C115 28E9               JR    Z,START         ; input a string again if end mark
                ;
C117 FE61               CP    'a'             ;
C119 3806               JR    C,DSPLAY        ;
C11B FE7B               CP    'z'+1           ; check and convert into capitals
C11D 3002               JR    NC,DSPLAY       ;
C11F CBAF               RES   5,A             ;
                ;
C121 CD5702     DSPLAY:CALL   0257H           ; dispalay a character
C124 18EC               JR    LOOP
                ;
                ; display message subroutine
                ;
                ;    inputs   : top address of message in register de
                ;    destroys : register a,f,d,e
```

```
                ;
C126 1A         DSPMSG:LD   A,(DE)
C127 A7                AND  A
C128 C8                RET  Z
C129 CD5702            CALL 0257H            ; display a character
C12C 13                INC  DE
C12D 18F7              JR   DSPMSG
                ;
                ; message area
                ;
C12F 0D0A6120 PROMPT:DEFB 0DH,0AH,'a string ? ',0
C133 73747269
C137 6E67203F
C13B 2000
C13D 63617069 ANSWER:DEFB 'capitals : ',0
C141 74616C73
C145 203A2000
                ;
C149                   END


*GC100

a string ? this is a dog.
capitals : THIS IS A DOG.
a string ? this is my dog.
capitals : THIS IS MY DOG.
a string ? my dog is in the box.
capitals : MY DOG IS IN THE BOX.
a string ? this is a cat.
capitals : THIS IS A CAT.
a string ? this is my cat.
capitals : THIS IS MY CAT.
a string ? my cat is in the box.
capitals : MY CAT IS IN THE BOX.
a string ? this is a ball.
capitals : THIS IS A BALL.
a string ? this is my ball.
capitals : THIS IS MY BALL.
a string ? my ball is on the table.
capitals : MY BALL IS ON THE TABLE.
a string ?

*
```

1F8BH BASICテキストの終了アドレス設定

アドレス	1F8BH
機　能	BASICテキストの終了アドレス設定
レジスタ	AFBCDEHL

解　説 レジスタHLに格納されたアドレスを，BASICテキスト・エリアの終了アドレス，すなわち変数エリアの先頭アドレスとして設定するため，各ワーク・エリアを適切な値でイニシアライズ後，プロンプト・メッセージ“Ok”を表示して，BASICのスクリーン・エディト・モードに入ります．

インタプリタでは，CMTインタフェースからBASICのプログラムを読み込んだ直後に，このルーチンを通過しています．

サンプル

```
                  ;
                  ; --- utility, memory to data version 3.0 ---
                  ;
                       ORG   0C100H
                  ;
C100 21E8FD            LD    HL,65000          ; starting line number of basic text
C103 22E8C1            LD    (LINENO),HL
                  ;
C106 3E5D              LD    A,']'
C108 CD5702            CALL  0257H             ; display a prompt mark
                  ;
C10B 210000            LD    HL,0              ;
C10E 54                LD    D,H               ; clear register de and register hl
C10F 5D                LD    E,L               ;
                  ;
C110 CD8CC1            CALL  INADDR            ; input top address of data block
C113 DA665C            JP    C,5C66H           ; exit if break or input error
C116 C2665C            JP    NZ,5C66H          ; exit if return key
                  ;
C119 EB                EX    DE,HL
C11A CD8CC1            CALL  INADDR            ; input end address of data block
C11D DA665C            JP    C,5C66H           ; exit if break or input error
C120 CA665C            JP    Z,5C66H           ; exit if comma key
                  ;
C123 CD9540            CALL  4095H             ; compare top address, end address
C126 DA665C            JP    C,5C66H           ; exit if end address > top address
                  ;
C129 22E6C1            LD    (ENDADR),HL       ; store end address
                  ;
C12C D5                PUSH  DE
C12D CD753D            CALL  3D75H             ; sense end address of basic text
C130 EB                EX    DE,HL
C131 D1                POP   DE
                  ;
C132 22E4C1       TOLINE:LD  (BASSTK),HL       ; store address of link-pointer
C135 23                INC   HL                ; skip it
C136 23                INC   HL                ; skip it
                  ;
C137 ED4BE8C1          LD    BC,(LINENO)       ;
C13B 71                LD    (HL),C            ;
C13C 23                INC   HL                ; write line number of text
C13D 70                LD    (HL),B            ;
C13E 23                INC   HL                ;
                  ;
```

```
C13F 03              INC  BC              ; increment line number
C140 ED43E8C1        LD   (LINENO),BC     ; store it
                ;
C144 3684            LD   (HL),84H        ; write intermediate code of 'data'
C146 23              INC  HL
                ;
C147 3620            LD   (HL),' '        ; write intermediate code of ' '
C149 23              INC  HL
                ;
C14A 0610            LD   B,16
C14C D5         NEWMEM:PUSH DE
C14D 1A              LD   A,(DE)          ; get byte data from memory
C14E CDCAC1          CALL ACCHEX          ; convert it into two characters
C151 72              LD   (HL),D          ; write msb byte of data
C152 23              INC  HL
C153 73              LD   (HL),E          ; write lsb byte of data
C154 23              INC  HL
C155 D1              POP  DE
                ;
C156 E5              PUSH HL
C157 2AE6C1          LD   HL,(ENDADR)     ; get end address of data block
C15A CD9540          CALL 4095H           ; check end address or not
C15D E1              POP  HL
C15E 2816            JR   Z,EOFILE        ; end of text if end of data
                ;
C160 13              INC  DE
C161 362C            LD   (HL),','        ; write intermediate code of ','
C163 23              INC  HL
C164 10E6            DJNZ NEWMEM          ; next data
                ;
C166 2B              DEC  HL
C167 3600            LD   (HL),0          ; write end of line mark
C169 23              INC  HL
                ;
C16A DD2AE4C1        LD   IX,(BASSTK)     ;
C16E DD7500          LD   (IX),L          ; write link-pointer of last line
C171 DD7401          LD   (IX+1),H        ;
                ;
C174 18BC            JR   TOLINE
                ;
C176 3600       EOFILE:LD (HL),0
C178 23              INC  HL
                ;
C179 DD2AE4C1        LD   IX,(BASSTK)     ;
C17D DD7500          LD   (IX),L          ; write link-pointer of last line
C180 DD7401          LD   (IX+1),H        ;
                ;
C183 3600            LD   (HL),0          ;
C185 23              INC  HL              ; write end of text mark
C186 3600            LD   (HL),0          ;
                ;
C188 23              INC  HL
C189 C38B1F          JP   1F8BH           ;*set fitting work area of basic
                ;
                ; input address subroutine
                ;
                ;    outputs  : input address in register hl
                ;               set carry flag if error or break
                ;               reset carry flag and reset zero flag if return
                ;               reset carry flag and set zero flag if comma
                ;    destroys : register a,f,h,l
                ;
C18C 210000     INADDR:LD HL,0
C18F CD750F     INADD1:CALL 0F75H         ; input a character
C192 CDCC4C          CALL 4CCCH           ; convert it into capital letter
C195 CD5702          CALL 0257H           ; echo-back it
                ;
C198 FE04            CP   04H             ; check stop key or not
C19A D8              RET  C               ; set cy if break
                ;
C19B FE0D            CP   0DH             ; check return key or not
C19D 2817            JR   Z,RESZ          ; reset cy and reset z if return key
                ;
C19F FE2C            CP   ','             ; check comma key or not
```

```
C1A1 C8              RET   Z                  ; reset cy and set z if comma key
                ;
C1A2 CDB8C1          CALL  CHKHEX             ; check hexa code or not
C1A5 D8              RET   C                  ; set cy if not hexa code
                ;
C1A6 FE3A            CP    '9'+1              ;
C1A8 3802            JR    C,INADD2           ;
C1AA C609            ADD   A,9                ;
C1AC E60F       INADD2:AND 0FH                ;
C1AE 29              ADD   HL,HL              ; convert into binary code
C1AF 29              ADD   HL,HL              ;   and store into register hl
C1B0 29              ADD   HL,HL              ;
C1B1 29              ADD   HL,HL              ;
C1B2 85              ADD   A,L                ;
C1B3 6F              LD    L,A                ;
                ;
C1B4 18D9            JR    INADD1             ; next digit
                ;
C1B6 3D         RESZ:  DEC  A                 ; reset zero flag
C1B7 C9              RET
                ;
                ; check hexa decimal code subroutine
                ;
                ;     inputs   : character code in accumulator
                ;     outputs  : reset carry flag if hexa code
                ;     destroys : register a,f
                ;
C1B8 FE30       CHKHEX:CP    '0'
C1BA D8              RET   C
C1BB FE47            CP    'F'+1
C1BD 3009            JR    NC,SETCY
C1BF FE3A            CP    '9'+1
C1C1 3803            JR    C,RESCY
C1C3 FE41            CP    'A'
C1C5 D8              RET   C
                ;
C1C6 A7         RESCY: AND  A            ; reset carry flag
C1C7 C9              RET
                ;
C1C8 37         SETCY: SCF               ; set carry flag
C1C9 C9              RET
                ;
                ; convert binary code into two hexa characters
                ;
                ;     inputs   : binary code in accumulator
                ;     outputs  : hexa character code in register d,e
                ;     destroys : a,f,d,e
                ;
C1CA F5         ACCHEX:PUSH AF
C1CB 0F              RRCA
C1CC 0F              RRCA
C1CD 0F              RRCA
C1CE 0F              RRCA
C1CF CDD9C1          CALL  HALF
C1D2 57              LD    D,A
C1D3 F1              POP   AF
C1D4 CDD9C1          CALL  HALF
C1D7 5F              LD    E,A
C1D8 C9              RET
                ;
C1D9 E60F       HALF:  AND  0FH
C1DB FE0A            CP    10
C1DD 3802            JR    C,NUMBER
C1DF C607            ADD   A,7
                ;
C1E1 C630       NUMBER:ADD  A,'0'
C1E3 C9              RET
                ;
                ; work area for work
                ;
C1E4            BASSTK:DS   2                 ; address of link-pointer
C1E6            ENDADR:DS   2                 ; end address of data block
C1E8            LINENO:DS   2                 ; line number of basic text
                ;
C1EA                 END
```

240FH　単精度型実数の減算

アドレス	240FH
機　　能	単精度型実数の減算
レジスタ	AFBCDEHL

解　説　レジスタBCDEに格納された単精度型の実数を被減数とし，フローティング・アキュムレータに格納された単精度型の実数を減数として，両者の間で減算を行い，演算結果をフローティング・アキュムレータに格納してもどります．

サンプル

```
                ;
                ; --- subtraction for single precision ---
                ;
                        ORG   0C100H
                ;
C100 2146C1     START:  LD    HL,PRMPT1
C103 CD3DC1             CALL  DSPMSG
C106 CD7E1B             CALL  1B7EH             ; screen editor
C109 DA665C             JP    C,5C66H           ; system monitor if stop key
C10C 23                 INC   HL
C10D CDBE2B             CALL  2BBEH             ; convert into binary code
C110 CDB327             CALL  27B3H             ; convert into single precision
C113 CDBD26             CALL  26BDH             ; move from facc to register bcde
C116 C5                 PUSH  BC
C117 D5                 PUSH  DE
                ;
C118 2156C1             LD    HL,PRMPT2
C11B CD3DC1             CALL  DSPMSG
C11E CD7E1B             CALL  1B7EH             ; screen editor
C121 23                 INC   HL
C122 CDBE2B             CALL  2BBEH             ; convert into binary code
C125 CDB327             CALL  27B3H             ; convert into single precision
                ;
C128 D1                 POP   DE
C129 C1                 POP   BC
C12A CD0F24             CALL  240FH             ;*subtraction for single precision
C12D CD222D             CALL  2D22H             ; convert into character code
C130 E5                 PUSH  HL
C131 2163C1             LD    HL,ANSWER
C134 CD3DC1             CALL  DSPMSG
C137 E1                 POP   HL
C138 CD3DC1             CALL  DSPMSG            ; display result
                ;
C13B 18C3               JR    START
                ;
                ; display message subroutine
                ;
C13D 7E         DSPMSG:LD     A,(HL)
C13E A7                 AND   A
C13F C8                 RET   Z
C140 CD5702             CALL  0257H             ; display a character
C143 23                 INC   HL
C144 18F7               JR    DSPMSG
                ;
                ; message area
                ;
C146 0D0A0A20   PRMPT1:DEFB 0DH,0AH,0AH,'      x!    ? ',0
C14A 20202020
C14E 78212020
C152 203F2000
```

```
C156 20202020 PRMPT2:DEFB '      y!   ? ',0
C15A 20792120
C15E 20203F20
C162 00
C163 7821202D ANSWER:DEFB 'x! - y! --> ',0
C167 20792120
C16B 2D2D3E20
C16F 00
                      ;
C170                  END


*GC100


      x!    ? 20
      y!    ? 30
x! - y! --> -10

      x!    ? 1000000
      y!    ? 1
x! - y! -->  999999

      x!    ? 12345678901234567890
      y!    ? 0
x! - y! -->  1.23457E+19

      x!    ? 0.5
      y!    ? 0.6
x! - y! --> -.1

      x!    ?

*
```

2412H　単精度型実数の加算

アドレス	2412H
機　　能	単精度型実数の加算
レジスタ	AFBCDEHL

解　説　レジスタＢＣＤＥに格納された単精度型の実数を被加数とし，フローティング・アキュムレータに格納された単精度型の実数を加数として，両者の間で加算を行い，演算結果をフローティング・アキュムレータに格納してもどります.

サンプル

```
                 ;
                 ; --- addition for single precision ---
                 ;
                        ORG  0C100H
                 ;
C100 2146C1      START: LD   HL,PRMPT1
C103 CD3DC1             CALL DSPMSG
C106 CD7E1B             CALL 1B7EH          ; screen editor
C109 DA665C             JP   C,5C66H        ; system monitor if stop key
C10C 23                 INC  HL
C10D CDBE2B             CALL 2BBEH          ; convert into binary code
C110 CDB327             CALL 27B3H          ; convert into single precision
C113 CDBD26             CALL 26BDH          ; move from facc to register bcde
C116 C5                 PUSH BC
C117 D5                 PUSH DE
                 ;
C118 2156C1             LD   HL,PRMPT2
C11B CD3DC1             CALL DSPMSG
C11E CD7E1B             CALL 1B7EH          ; screen editor
C121 23                 INC  HL
C122 CDBE2B             CALL 2BBEH          ; convert into binary code
C125 CDB327             CALL 27B3H          ; convert into single precision
                 ;
C128 D1                 POP  DE
C129 C1                 POP  BC
C12A CD1224             CALL 2412H          ;*addition for single precision
C12D CD222D             CALL 2D22H          ; convert into character code
C130 E5                 PUSH HL
C131 2163C1             LD   HL,ANSWER
C134 CD3DC1             CALL DSPMSG
C137 E1                 POP  HL
C138 CD3DC1             CALL DSPMSG         ; display result
                 ;
C13B 18C3               JR   START
                 ;
                 ; display message subroutine
                 ;
C13D 7E          DSPMSG:LD   A,(HL)
C13E A7                 AND  A
C13F C8                 RET  Z
C140 CD5702             CALL 0257H          ; display a character
C143 23                 INC  HL
C144 18F7               JR   DSPMSG
                 ;
                 ; message area
                 ;
C146 0D0A0A20    PRMPT1:DEFB 0DH,0AH,0AH,'      x!    ? ',0
C14A 20202020
C14E 78212020
C152 203F2000
```

```
C156 20202020 PRMPT2:DEFB '      y!    ? ',0
C15A 20792120
C15E 20203F20
C162 00
C163 7821202B ANSWER:DEFB 'x! + y! --> ',0
C167 20792120
C16B 2D2D3E20
C16F 00
                       ;
C170                   END


*GC100


     x!    ? 0.123
     y!    ? 0.007
x! + y! -->  .13

     x!    ? 20
     y!    ? -50
x! + y! --> -30

     x!    ? 123.456
     y!    ? 0.0000001
x! + y! -->  123.456

     x!    ? abc
     y!    ? def
x! + y! -->  0

     x!    ?

*
```

2503H　自然対数の計算

アドレス	2503H
機　能	自然対数の計算
レジスタ	AFBCDEHL

解　説　フローティング・アキュムレータに格納された単精度型実数の，自然対数（e を底とした対数）の値を求め，同じくフローティング・アキュムレータに与えてもどります．

サンプル

```
                ;
                ; --- log function test progtam ---
                ;
                        ORG   0C100H
                ;
C100 213AC1     START: LD    HL,PROMPT
C103 CD31C1            CALL  DSPMSG
                ;
C106 CD7E1B            CALL  1B7EH            ; screen editor
C109 DA665C            JP    C,5C66H          ; system monitor if stop key
                ;
C10C 23                INC   HL
C10D E5                PUSH  HL
                ;
C10E CDBE2B            CALL  2BBEH            ; convert into binary code
C111 CDB327            CALL  27B3H            ; convert into single precision
C114 CD0325            CALL  2503H            ;*log function
C117 CD222D            CALL  2D22H            ; convert into character code
                ;
C11A EB                EX    DE,HL            ; top address of result in reg-de
                ;
C11B 2148C1            LD    HL,ANS1
C11E CD31C1            CALL  DSPMSG
                ;
C121 E1                POP   HL               ; top address of a input number
C122 CD31C1            CALL  DSPMSG           ; display a input number again
                ;
C125 2154C1            LD    HL,ANS2
C128 CD31C1            CALL  DSPMSG
                ;
C12B EB                EX    DE,HL            ; top address of result in reg-hl
C12C CD31C1            CALL  DSPMSG           ; display result
                ;
C12F 18CF              JR    START
                ;
                ; display message subroutine
                ;
C131 7E         DSPMSG:LD    A,(HL)
C132 A7                AND   A
C133 C8                RET   Z
C134 CD5702            CALL  0257H            ; display a character
C137 23                INC   HL
C138 18F7              JR    DSPMSG
                ;
                ; message area
                ;
C13A 0D0A6120   PROMPT:DEFB 0DH,0AH,'a number ? ',0
C13E 6E756D62
C142 6572203F
C146 2000
C148 20202020   ANS1:  DEFB '     log ( ',0
```

```
C14C 206C6F67
C150 20282000
C154 20292069 ANS2:  DEFB ' ) is equal to ',0
C158 73206571
C15C 75616C20
C160 746F2000
              ;
C164                 END
```

```
*GC100

a number ? 0.001
     log ( 0.001 ) is equal to -6.90776
a number ? 0.01
     log ( 0.01 ) is equal to -4.60517
a number ? 0.1
     log ( 0.1 ) is equal to -2.30259
a number ? 1
     log ( 1 ) is equal to  0
a number ? 10
     log ( 10 ) is equal to  2.30259
a number ? 100
     log ( 100 ) is equal to  4.60517
a number ? 10000
     log ( 10000 ) is equal to  9.21034
a number ? 100000000
     log ( 100000000 ) is equal to  18.4207
a number ? -100
Illegal function call
Ok
```

2541H　単精度型実数の乗算

アドレス	2541H
機　　能	単精度型実数の乗算
レジスタ	AFBCDEHL

解　　説　レジスタBCDEに格納された単精度型の実数を被乗数とし，フローティング・アキュムレータに格納された単精度型の実数を乗数として，両者の間で乗算を行い，演算結果をフローティング・アキュムレータに格納してもどります．

サンプル

```
                ;
                ; --- multiplication for single precision ---
                ;
                        ORG   0C100H
                ;
C100 2146C1     START:  LD    HL,PRMPT1
C103 CD3DC1             CALL  DSPMSG
C106 CD7E1B             CALL  1B7EH            ; screen editor
C109 DA665C             JP    C,5C66H          ; system monitor if stop key
C10C 23                 INC   HL
C10D CDBE2B             CALL  2BBEH            ; convert into binary code
C110 CDB327             CALL  27B3H            ; convert into single precision
C113 CDBD26             CALL  26BDH            ; move from facc to register bcde
C116 C5                 PUSH  BC
C117 D5                 PUSH  DE
                ;
C118 2156C1             LD    HL,PRMPT2
C11B CD3DC1             CALL  DSPMSG
C11E CD7E1B             CALL  1B7EH            ; screen editor
C121 23                 INC   HL
C122 CDBE2B             CALL  2BBEH            ; convert into binary code
C125 CDB327             CALL  27B3H            ; convert into single precision
                ;
C128 D1                 POP   DE
C129 C1                 POP   BC
C12A CD4125             CALL  2541H            ;*multiplication for single
C12D CD222D             CALL  2D22H            ; convert into character code
C130 E5                 PUSH  HL
C131 2163C1             LD    HL,ANSWER
C134 CD3DC1             CALL  DSPMSG
C137 E1                 POP   HL
C138 CD3DC1             CALL  DSPMSG           ; display result
                ;
C13B 18C3               JR    START
                ;
                ; display message subroutine
                ;
C13D 7E         DSPMSG:LD     A,(HL)
C13E A7                 AND   A
C13F C8                 RET   Z
C140 CD5702             CALL  0257H            ; display a character
C143 23                 INC   HL
C144 18F7               JR    DSPMSG
                ;
                ; message area
                ;
C146 0D0A0A20   PRMPT1:DEFB 0DH,0AH,0AH,'      x!   ? ',0
C14A 20202020
C14E 78212020
```

```
C152 203F2000
C156 20202020 PRMPT2:DEFB '      y!   ? ',0
C15A 20792120
C15E 20203F20
C162 00
C163 7821202A ANSWER:DEFB 'x! * y! --> ',0
C167 20792120
C16B 2D2D3E20
C16F 00
                      ;
C170                  END


*GC100


     x!   ? 10000
     y!   ? 100
x! * y! -->  1E+06

     x!   ? 1
     y!   ? 1
x! * y! -->  1

     x!   ? 2
     y!   ? 2
x! * y! -->  4

     x!   ? 100
     y!   ? 123
x! * y! -->  12300

     x!   ?

*
```

259CH　単精度型実数の除算

アドレス	259CH
機　能	単精度型実数の除算
レジスタ	AFBCDEHL

解　説　レジスタＢＣＤＥに格納された単精度型の実数を被除数とし，フローティング・アキュムレータに格納された単精度型の実数を除数として，両者の間で除算を行い，演算結果をフローティング・アキュムレータに格納してもどります．

フローティング・アキュムレータに与える除数が0（ゼロ）であった場合には，"Division by zero" エラーが発生し，インタプリタの制御下に入ります．

サンプル

```
               ;
               ; --- division for single precision ---
               ;
                       ORG   0C100H
               ;
C100 2146C1    START:  LD    HL,PRMPT1
C103 CD3DC1            CALL  DSPMSG
C106 CD7E1B            CALL  1B7EH          ; screen editor
C109 DA665C            JP    C,5C66H        ; system monitor if stop key
C10C 23                INC   HL
C10D CDBE2B            CALL  2BBEH          ; convert into binary code
C110 CDB327            CALL  27B3H          ; convert into single precision
C113 CDBD26            CALL  26BDH          ; move from facc to register bcde
C116 C5                PUSH  BC
C117 D5                PUSH  DE
               ;
C118 2156C1            LD    HL,PRMPT2
C11B CD3DC1            CALL  DSPMSG
C11E CD7E1B            CALL  1B7EH          ; screen editor
C121 23                INC   HL
C122 CDBE2B            CALL  2BBEH          ; convert into binary code
C125 CDB327            CALL  27B3H          ; convert into single precision
               ;
C128 D1                POP   DE
C129 C1                POP   BC
C12A CD9C25            CALL  259CH          ;*division for single precision
C12D CD222D            CALL  2D22H          ; convert into character code
C130 E5                PUSH  HL
C131 2163C1            LD    HL,ANSWER
C134 CD3DC1            CALL  DSPMSG
C137 E1                POP   HL
C138 CD3DC1            CALL  DSPMSG         ; display result
               ;
C13B 18C3              JR    START
               ;
               ; display message subroutine
               ;
C13D 7E        DSPMSG:LD     A,(HL)
C13E A7                AND   A
C13F C8                RET   Z
C140 CD5702            CALL  0257H          ; display a character
C143 23                INC   HL
C144 18F7              JR    DSPMSG
               ;
```

```
                ; massage area
                ;
C146 0D0A0A20 PRMPT1:DEFB 0DH,0AH,0AH,'      x!    ? ',0
C14A 20202020
C14E 78212020
C152 203F2000
C156 20202020 PRMPT2:DEFB '      y!    ? ',0
C15A 20792120
C15E 20203F20
C162 00
C163 7821202F ANSWER:DEFB 'x! / y! --> ',0
C167 20792120
C16B 2D2D3E20
C16F 00
                ;
C170            END


*GC100


     x!    ? 100000
     y!    ? 10
x! / y! -->  10000

     x!    ? 10
     y!    ? -3
x! / y! --> -3.33333

     x!    ? 20
     y!    ? 6
x! / y! -->  3.33333

     x!    ? -5
     y!    ? -5
x! / y! -->  1

     x!    ? 5
     y!    ? abc
Division by zero
Ok
```

２６７１Ｈ　絶対値の計算

アドレス	２６７１Ｈ
機　　能	絶対値の計算
レジスタ	ＡＦＢＣＤＥＨＬ

解　　説　フローティング・アキュムレータに格納された数値データの絶対値を求め，同じくフローティング・アキュムレータに与えてもどります．

演算結果は，フローティング・アキュムレータに格納された数値データが倍精度実数型の場合には倍精度実数型に，それ以外の場合には単精度実数型になります．

サンプル

```
                ;
                ; --- abs function test progtam ---
                ;
                        ORG   0C100H
                ;
C100 2137C1     START:  LD    HL,PROMPT
C103 CD2EC1             CALL  DSPMSG
                ;
C106 CD7E1B             CALL  1B7EH           ; screen editor
C109 DA665C             JP    C,5C66H         ; system monitor if stop key
                ;
C10C 23                 INC   HL
C10D E5                 PUSH  HL
                ;
C10E CDBE2B             CALL  2BBEH           ; convert into binary code
C111 CD7126             CALL  2671H           ;*abs function
C114 CD222D             CALL  2D22H           ; convert into character code
                ;
C117 EB                 EX    DE,HL           ; top address of result in reg-de
                ;
C118 2145C1             LD    HL,ANS1
C11B CD2EC1             CALL  DSPMSG
                ;
C11E E1                 POP   HL              ; top address of a input number
C11F CD2EC1             CALL  DSPMSG          ; display a input number again
                ;
C122 2151C1             LD    HL,ANS2
C125 CD2EC1             CALL  DSPMSG
                ;
C128 EB                 EX    DE,HL           ; top address of result in reg-hl
C129 CD2EC1             CALL  DSPMSG          ; display result
                ;
C12C 18D2               JR    START
                ;
                ; display message subroutine
                ;
C12E 7E         DSPMSG: LD    A,(HL)
C12F A7                 AND   A
C130 C8                 RET   Z
C131 CD5702             CALL  0257H           ; display a character
C134 23                 INC   HL
C135 18F7               JR    DSPMSG
                ;
                ; message area
                ;
C137 0D0A6120   PROMPT: DEFB  0DH,0AH,'a number ? ',0
C13B 6E756D62
```

```
C13F 6572203F
C143 2000
C145 20202020 ANS1:  DEFB '     abs ( ',0
C149 20616273
C14D 20282000
C151 20292069 ANS2:  DEFB ' ) is equal to ',0
C155 73206571
C159 75616C20
C15D 746F2000
              ;
C161                 END
```

```
*GC100

a number ? 3
     abs ( 3 ) is equal to  3
a number ? -3
     abs ( -3 ) is equal to  3
a number ? 3%
     abs ( 3% ) is equal to  3
a number ? -3%
     abs ( -3% ) is equal to  3
a number ? 3.3
     abs ( 3.3 ) is equal to  3.3
a number ? -3.3
     abs ( -3.3 ) is equal to  3.3
a number ? 3333333333333333
     abs ( 3333333333333333 ) is equal to  3333333333333333
a number ? -3333333333333333
     abs ( -3333333333333333 ) is equal to  3333333333333333
a number ? &h11
     abs ( &h11 ) is equal to  17
a number ? &hfffe
     abs ( &hfffe ) is equal to  2
a number ?

*
```

２６７ＥＨ　実数の符号反転

アドレス	２６７ＥＨ
機　　能	実数の符号反転
レジスタ	ＡＦ－－－－ＨＬ

解　　説　フローティング・アキュムレータに格納されている単精度型実数または倍精度型実数の符号を反転（ＮＡＧＡＴＩＯＮ）してもどります．

具体的には，フローティング・アキュムレータの仮数部最上位バイト（Ｆ０ＡＡＨ番地）の最上位ビットを反転することによって符号の反転を実現しており，ＥＦ４５Ｈ番地のチェック等エラー処理は全く行っていませんから，フローティング・アキュムレータのデータは必ず実数型でなければなりません．

サンプル

```
                ;
                ; --- negation for the real number ---
                ;
                        ORG   0C100H
                ;
C100 2134C1     START:  LD    HL,PROMPT
C103 CD2BC1             CALL  DSPMSG
C106 CD7E1B             CALL  1B7EH            ; screen editor
C109 DA665C             JP    C,5C66H          ; system monitor if stop key
C10C 23                 INC   HL
C10D CDBE2B             CALL  2BBEH            ; convert into binary code
                ;
C110 3A45EF             LD    A,(0EF45H)       ;
C113 FE02               CP    2                ; convert into the real number
C115 CCD027             CALL  Z,27D0H          ;
                ;
C118 CD7E26             CALL  267EH            ;*negation for the real number
                ;
C11B CD222D             CALL  2D22H            ; convert into character code
C11E E5                 PUSH  HL
C11F 214CC1             LD    HL,ANSWER
C122 CD2BC1             CALL  DSPMSG
C125 E1                 POP   HL
C126 CD2BC1             CALL  DSPMSG           ; display result
                ;
C129 18D5               JR    START
                ;
                ; display message subroutine
                ;
                ;    destroys : register a,h,l
                ;
C12B 7E         DSPMSG:LD     A,(HL)
C12C A7                 AND   A
C12D C8                 RET   Z
C12E CD5702             CALL  0257H            ; display a character
C131 23                 INC   HL
C132 18F7               JR    DSPMSG
                ;
                ; message area
                ;
C134 0D0A0A61   PROMPT:DEFB 0DH,0AH,0AH,'a real number x   ? ',0
C138 20726561
C13C 6C206E75
C140 6D626572
C144 20782020
```

```
C148 203F2000
C14C 20202020 ANSWER:DEFB '                 -x --> ',0
C150 20202020
C154 20202020
C158 202D7820
C15C 2D2D3E20
C160 00
                ;
C161                    END


*GC100


a real number x    ? 123456789
              -x --> -123456789

a real number x    ? -123456789
              -x -->  123456789

a real number x    ? -1.5
              -x -->  1.5

a real number x    ? &h11
              -x --> -17

a real number x    ? &Hfffe
              -x -->  2

a real number x    ? -3%
              -x -->  3

a real number x    ?

*
```

２６８６H　符号の調査

アドレス	２６８６H
機　能	符号の調査
レジスタ	AF－－－－HL

解　説　フローティング・アキュムレータに格納された数値データが，正の場合には１を，０（ゼロ）の場合には０を，負の場合には－１を，それぞれフローティング・アキュムレータに整数形式で格納し，ＥＦ４５Ｈ番地にも整数型を示す０２Ｈを与えます．

ＥＦ４５Ｈ番地に０３Ｈが入っていた場合には，“Type mismatch”エラーが発生し，インタプリタの制御下に入ります．

サンプル

```
                ;
                ; --- sgn function test progtam ---
                ;
                        ORG   0C100H
                ;
C100 2154C1     START: LD    HL,PROMPT
C103 CD4BC1            CALL  DSPMSG
                ;
C106 CD7E1B            CALL  1B7EH          ; screen editor
C109 DA665C            JP    C,5C66H        ; system monitor if stop key
                ;
C10C 23                INC   HL
C10D E5                PUSH  HL
                ;
C10E CDBE2B            CALL  2BBEH          ; convert into binary code
C111 CD8626            CALL  2686H          ;*sgn function
                ;
C114 3AA8F0            LD    A,(0F0A8H)
C117 08                EX    AF,AF'
                ;
C118 CD222D            CALL  2D22H          ; convert into character code
                ;
C11B EB                EX    DE,HL          ; top address of result in reg-de
                ;
C11C 2162C1            LD    HL,ANS1
C11F CD4BC1            CALL  DSPMSG
                ;
C122 E1                POP   HL             ; top address of a input number
C123 CD4BC1            CALL  DSPMSG         ; display a input number again
                ;
C126 216EC1            LD    HL,ANS2
C129 CD4BC1            CALL  DSPMSG
                ;
C12C EB                EX    DE,HL          ; top address of result in reg-hl
C12D CD4BC1            CALL  DSPMSG         ; display result
                ;
C130 08                EX    AF,AF'
                ;
C131 FE01              CP    01H
C133 2005              JR    NZ,NOPOSI
                ;
C135 217EC1            LD    HL,POSI
C138 180C              JR    DSPLAY
                ;
```

```
C13A FE00      NOPOSI:CP    00H
C13C 2005             JR    NZ,NOZERO
               ;
C13E 219CC1           LD    HL,ZERO
C141 1803             JR    DSPLAY
               ;
C143 21B4C1    NOZERO:LD    HL,NEGA
               ;
C146 CD4BC1    DSPLAY:CALL  DSPMSG
               ;
C149 18B5             JR    START
               ;
               ; display message subroutine
               ;
C14B 7E        DSPMSG:LD    A,(HL)
C14C A7               AND   A
C14D C8               RET   Z
C14E CD5702           CALL  0257H           ; display a character
C151 23               INC   HL
C152 18F7             JR    DSPMSG
               ;
               ; message area
               ;
C154 0D0A6120 PROMPT:DEFB 0DH,0AH,'a number ? ',0
C158 6E756D62
C15C 6572203F
C160 2000
C162 20202020 ANS1:  DEFB '     abs ( ',0
C166 20616273
C16A 20282000
C16E 20292069 ANS2:  DEFB ' ) is equal to ',0
C172 73206571
C176 75616C20
C17A 746F2000
C17E 20202020 POSI:  DEFB '     means positive number !!',0
C182 206D6561
C186 6E732070
C18A 6F736974
C18E 69766520
C192 6E756D62
C196 65722021
C19A 2100
C19C 20202020 ZERO:  DEFB '     means just zero !!',0
C1A0 206D6561
C1A4 6E73206A
C1A8 75737420
C1AC 7A65726F
C1B0 20212100
C1B4 20202020 NEGA:  DEFB '     means negative number !!',0
C1B8 206D6561
C1BC 6E73206E
C1C0 65676174
C1C4 69766520
C1C8 6E756D62
C1CC 65722021
C1D0 2100
               ;
C1D2                 END
```

```
*GC100

a number ? 1
     abs ( 1 ) is equal to  1     means positive number !!
a number ? 0
     abs ( 0 ) is equal to  0     means just zero !!
a number ? -1
     abs ( -1 ) is equal to -1     means negative number !!
a number ? -0.0000000000000001
     abs ( -0.0000000000000001 ) is equal to -1     means negative number !!
a number ? z80
     abs (z80) is equal to 0     means just zero !!
a number ?
*
```

２６８９Ｈ　８ビット整数の格納

アドレス	２６８９Ｈ
機　能	８ビット整数の格納
レジスタ	ＡＦ－－－－ＨＬ

解　説　アキュムレータに格納された有符号の８ビット整数（－１２８～１２７)を，有符号１６ビット整数に変換して，フローティング･アキュムレータ(Ｆ０Ａ８Ｈ～Ｆ０Ａ９Ｈ番地）に格納し，ＥＦ４５Ｈ番地にも整数型を示す０２Ｈを格納してもどります．

サンプル

```
                ;
                ; --- signed numbers with a byte ---
                ;
                        ORG   0C100H
                ;
C100 3E80               LD    A,80H
                ;
C102 F5         LOOP:   PUSH AF
                ;
C103 F5                 PUSH AF
C104 CD8926             CALL 2689H              ;*store accumulator into facc
C107 F1                 POP   AF
                ;
C108 CD2BC1             CALL DSPACC             ; display accumulaor hexa decimal
                ;
C10B 3E80               LD    A,80H             ;
C10D 0605               LD    B,5               ; appoint format
C10F 0E00               LD    C,0               ;
                ;
C111 CD232D             CALL 2D23H              ; convert into character code
C114 CD42C1             CALL DSPMSG             ; display the number decimal
                ;
C117 3E20               LD    A,' '
C119 CD5702             CALL 0257H
C11C CD5702             CALL 0257H
C11F CD5702             CALL 0257H
                ;
C122 F1                 POP   AF
                ;
C123 3C                 INC   A
C124 FE80               CP    80H
C126 20DA               JR    NZ,LOOP
                ;
C128 C3665C            .JP    5C66H             ; jump to system monitor
                ;
                ; display accumulator hexa decimal subroutine
                ;
                ;    destroys : register a,f,b
                ;
C12B 47         DSPACC:LD     B,A
C12C 0F                 RRCA
C12D 0F                 RRCA
C12E 0F                 RRCA
C12F 0F                 RRCA
C130 CD34C1             CALL HALF
C133 78                 LD    A,B
                ;
C134 E60F       HALF:   AND   0FH
C136 FE0A               CP    10
C138 3802               JR    C,NUMBER
```

```
C13A C607              ADD  A,7
                  ;
C13C C630         NUMBER:ADD  A,30H
C13E CD5702            CALL 0257H          ; display a character
C141 C9                RET
                  ;
                  ; display message subroutine
                  ;
                  ;    destroys : register a,f,h,l
                  ;
C142 7E           DSPMSG:LD   A,(HL)
C143 A7                AND  A
C144 C8                RET  Z
C145 CD5702            CALL 0257H          ; display a character
C148 23                INC  HL
C149 18F7              JR   DSPMSG
                  ;
C14B                   END
```

```
*GC100
80 -128   81 -127   82 -126   83 -125   84 -124   85 -123   86 -122   87 -121
88 -120   89 -119   8A -118   8B -117   8C -116   8D -115   8E -114   8F -113
90 -112   91 -111   92 -110   93 -109   94 -108   95 -107   96 -106   97 -105
98 -104   99 -103   9A -102   9B -101   9C -100   9D  -99   9E  -98   9F  -97
A0  -96   A1  -95   A2  -94   A3  -93   A4  -92   A5  -91   A6  -90   A7  -89
A8  -88   A9  -87   AA  -86   AB  -85   AC  -84   AD  -83   AE  -82   AF  -81
B0  -80   B1  -79   B2  -78   B3  -77   B4  -76   B5  -75   B6  -74   B7  -73
B8  -72   B9  -71   BA  -70   BB  -69   BC  -68   BD  -67   BE  -66   BF  -65
C0  -64   C1  -63   C2  -62   C3  -61   C4  -60   C5  -59   C6  -58   C7  -57
C8  -56   C9  -55   CA  -54   CB  -53   CC  -52   CD  -51   CE  -50   CF  -49
D0  -48   D1  -47   D2  -46   D3  -45   D4  -44   D5  -43   D6  -42   D7  -41
D8  -40   D9  -39   DA  -38   DB  -37   DC  -36   DD  -35   DE  -34   DF  -33
E0  -32   E1  -31   E2  -30   E3  -29   E4  -28   E5  -27   E6  -26   E7  -25
E8  -24   E9  -23   EA  -22   EB  -21   EC  -20   ED  -19   EE  -18   EF  -17
F0  -16   F1  -15   F2  -14   F3  -13   F4  -12   F5  -11   F6  -10   F7   -9
F8   -8   F9   -7   FA   -6   FB   -5   FC   -4   FD   -3   FE   -2   FF   -1
00    0   01    1   02    2   03    3   04    4   05    5   06    6   07    7
08    8   09    9   0A   10   0B   11   0C   12   0D   13   0E   14   0F   15
10   16   11   17   12   18   13   19   14   20   15   21   16   22   17   23
18   24   19   25   1A   26   1B   27   1C   28   1D   29   1E   30   1F   31
20   32   21   33   22   34   23   35   24   36   25   37   26   38   27   39
28   40   29   41   2A   42   2B   43   2C   44   2D   45   2E   46   2F   47
30   48   31   49   32   50   33   51   34   52   35   53   36   54   37   55
38   56   39   57   3A   58   3B   59   3C   60   3D   61   3E   62   3F   63
40   64   41   65   42   66   43   67   44   68   45   69   46   70   47   71
48   72   49   73   4A   74   4B   75   4C   76   4D   77   4E   78   4F   79
50   80   51   81   52   82   53   83   54   84   55   85   56   86   57   87
58   88   59   89   5A   90   5B   91   5C   92   5D   93   5E   94   5F   95
60   96   61   97   62   98   63   99   64  100   65  101   66  102   67  103
68  104   69  105   6A  106   6B  107   6C  108   6D  109   6E  110   6F  111
70  112   71  113   72  114   73  115   74  116   75  117   76  118   77  119
78  120   79  121   7A  122   7B  123   7C  124   7D  125   7E  126   7F  127
*
```

２６ＡＦＨ　単精度型実数の移動１

アドレス	２６ＡＦＨ
機　能	単精度型実数の移動１
レジスタ	ーーＢＣＤＥＨＬ

解　説　レジスタＨＬで指定するアドレスから４バイトに渡って格納された単精度型の実数を，レジスタＢＣＤＥおよびフローティング・アキュムレータに格納してもどります．

２６Ｂ２Ｈ　単精度型実数の移動２

アドレス	２６Ｂ２Ｈ
機　能	単精度型実数の移動２
レジスタ	ーーーーＤＥーー

解　説　レジスタＢＣＤＥに格納された単精度型の実数をフローティング・アキュムレータに格納し，ＥＦ４５Ｈ番地にも単精度実数型を示す０４Ｈを入れてもどります．

２６ＢＤＨ　単精度型実数の移動３

アドレス	２６ＢＤＨ
機　能	単精度型実数の移動３
レジスタ	ーーＢＣＤＥＨＬ

解　説　フローティング・アキュムレータに格納された単精度型の実数を，レジスタＢＣＤＥに格納してもどります．

２６Ｃ０Ｈ　単精度型実数の移動４

アドレス	２６Ｃ０Ｈ
機　　能	単精度型実数の移動４
レジスタ	－－ＢＣＤＥＨＬ

解　　説　レジスタＨＬで指定するアドレスから４バイトに渡って格納された単精度型の実数を，レジスタＢＣＤＥに格納してもどります．

２６Ｃ９Ｈ　単精度型実数の移動５

アドレス	２６Ｃ９Ｈ
機　　能	単精度型実数の移動５
レジスタ	Ａ－ＢＣＤＥＨＬ

解　　説　フローティング・アキュームレタに格納された単精度型の実数を，レジスタＨＬで指定するアドレスから４バイトに渡るメモリに格納してもどります．

２６Ｄ５Ｈ　２５６バイト以内のブロック転送

アドレス	２６Ｄ５Ｈ
機　　能	２５６バイト以内のブロック転送
レジスタ	ＡＦＢ－ＤＥＨＬ

解　　説　レジスタＤＥで指定するアドレスからレジスタＨＬで指定するアドレスに，レジスタＢで指定するバイト量のデータをブロック転送しますが，レジスタＢの内容が００Ｈの場合には２５６バイトを指定したことになります．

２７０ＣＨ　単精度型実数の比較

アドレス	２７０ＣＨ
機　　能	単精度型実数の比較
レジスタ	ＡＦ－－－－ＨＬ

解　　説　レジスタＢＣＤＥに格納された単精度型の実数を第１パラメータとし，フローティング・アキュムレータに格納された単精度型の実数を第２パラメータとして，両者の比較結果をアキュムレータに格納し，ゼロ・フラグ（Ｚ）およびキャリ・フラグ（ＣＹ）に影響を与えてもどります．

比較結果とアキュムレータに与える数値の関係は次のとおりです．

第１パラメータ＜第２パラメータの場合，比較結果に０１Ｈを与えます．

第１パラメータ＝第２パラメータの場合，比較結果に００Ｈを与えます．

第１パラメータ＞第２パラメータの場合，比較結果にＦＦＨを与えます．

サンプル

```
                 ;
                 ; --- compare for single precision ---
                 ;
                        ORG   0C100H
                 ;
C100 214DC1      START: LD    HL,PRMPT1
C103 CD44C1             CALL  DSPMSG
C106 CD7E1B             CALL  1B7EH             ; screen editor
C109 DA665C             JP    C,5C66H           ; jump to system monitor if stop key
C10C 23                 INC   HL
C10D CDBE2B             CALL  2BBEH             ; convert into binary code
C110 CDB327             CALL  27B3H             ; convert into single precision
                 ;
C113 CDBD26             CALL  26BDH             ;
C116 C5                 PUSH  BC                ; push facc onto stack
C117 D5                 PUSH  DE                ;
                 ;
C118 216AC1             LD    HL,PRMPT2
C11B CD44C1             CALL  DSPMSG
C11E CD7E1B             CALL  1B7EH             ; screen editor
C121 23                 INC   HL
C122 CDBE2B             CALL  2BBEH             ; convert into binary code
C125 CDB327             CALL  27B3H             ; convert into single precision
                 ;
C128 D1                 POP   DE
C129 C1                 POP   BC
                 ;
C12A CD0C27             CALL  270CH             ;*compare for single precision
                 ;
C12D 2005               JR    NZ,NEQUAL
C12F 2184C1             LD    HL,ANS1           ; x! equal y!
C132 180B               JR    DSPLAY
                 ;
C134 D23CC1      NEQUAL:JP    NC,LESS
C137 2191C1             LD    HL,ANS2           ; x! greater than y!
C13A 1803               JR    DSPLAY
                 ;
C13C 21A4C1      LESS:  LD    HL,ANS3           ; x! less than y!
                 ;
C13F CD44C1      DSPLAY:CALL  DSPMSG            ; display result
```

```
C142 18BC              JR    START
                ;
                ; display message subroutine
                ;
                ;    destroys : register a,f,h,l
                ;
C144 7E         DSPMSG:LD    A,(HL)
C145 A7                AND   A
C146 C8                RET   Z
C147 CD5702            CALL  0257H           ; display a character
C14A 23                INC   HL
C14B 18F7              JR    DSPMSG
                ;
                ; message area
                ;
C14D 0D0A0A73 PRMPT1:DEFB 0DH,0AH,0AH,'single precision of x! ? ',0
C151 696E676C
C155 65207072
C159 65636973
C15D 696F6E20
C161 6F662078
C165 21203F20
C169 00
C16A 73696E67 PRMPT2:DEFB 'single precision of x! ? ',0
C16E 6C652070
C172 72656369
C176 73696F6E
C17A 206F6620
C17E 7921203F
C182 2000
C184 78212065 ANS1:  DEFB 'x! equal  y!',0
C188 7175616C
C18C 20207921
C190 00
C191 78212067 ANS2:  DEFB 'x! greater than y!',0
C195 72656174
C199 65722074
C19D 68616E20
C1A1 792100
C1A4 7821206C ANS3:  DEFB 'x! less than y!',0
C1A8 65737320
C1AC 7468616E
C1B0 20792100
                ;
C1B4                   END


*GC100


single precision of x! ? 123456.789
single precision of y! ? 1234567.89
x! less than y!

single precision of x! ? 123456789
single precision of y! ? 123456788
x! equal  y!

single precision of x! ? 1.11111
single precision of y! ? 1.111111
x! less than y!

single precision of x! ? 1.11111
single precision of y! ? 1.11111000000001
x! equal  y!

single precision of x! ?

*
```

２７３９Ｈ　整数の比較

アドレス　２７３９Ｈ

機　　能　整数の比較

レジスタ　ＡＦ－－－－－－

解　　説　レジスタＤＥに格納された有符号１６ビットの整数を第１パラメータとし，レジスタＨＬに格納された有符号１６ビットの整数を第２パラメータとして，両者の比較結果をアキュムレータに格納し，ゼロ・フラグ（Ｚ）およびキャリ・フラグ（ＣＹ）に影響を与えてもどります．

比較結果とアキュムレータに与える数値の関係は次のとおりです．

第１パラメータ＜第２パラメータの場合，比較結果に０１Ｈを与えます．

第１パラメータ＝第２パラメータの場合，比較結果に００Ｈを与えます．

第１パラメータ＞第２パラメータの場合，比較結果にＦＦＨを与えます．

サンプル

```
                ;
                ; --- compare for integer ---
                ;
                        ORG   0C100H
                ;
C100 214FC1     START: LD    HL,PRMPT1
C103 CD46C1            CALL  DSPMSG
C106 CD7E1B            CALL  1B7EH          ; screen editor
C109 DA665C            JP    C,5C66H        ; jump to system monitor if stop key
C10C 23                INC   HL
C10D CDBE2B            CALL  2BBEH          ; convert into binary code
C110 CD7F27            CALL  277FH          ; convert into integer
                ;
C113 ED5BA8F0          LD    DE,(0F0A8H)    ; store 1st argument into reg-de
C117 D5                PUSH  DE
                ;
C118 2163C1            LD    HL,PRMPT2
C11B CD46C1            CALL  DSPMSG
C11E CD7E1B            CALL  1B7EH          ; screen editor
C121 23                INC   HL
C122 CDBE2B            CALL  2BBEH          ; convert into binary code
C125 CD7F27            CALL  277FH          ; convert into integer
C128 2AA8F0            LD    HL,(0F0A8H)    ; set 2nd argument into reg-hl
                ;
C12B D1                POP   DE             ; set 1st argument into reg-de
                ;
C12C CD3927            CALL  2739H          ;*compare for integer
                ;
C12F 2005              JR    NZ,NEQUAL
C131 2174C1            LD    HL,ANS1        ; x! equal y!
C134 180B              JR    DSPLAY
                ;
C136 D23EC1     NEQUAL:JP    NC,LESS
C139 2180C1            LD    HL,ANS2        ; x! greater than y!
C13C 1803              JR    DSPLAY
                ;
C13E 2193C1     LESS:  LD    HL,ANS3        ; x! less than y!
                ;
C141 CD46C1     DSPLAY:CALL  DSPMSG         ; display result
C144 18BA              JR    START
```

```
                ;
                ; display message subroutine
                ;
                ;    destroys : register a,f,h,l
                ;
C146 7E         DSPMSG:LD   A,(HL)
C147 A7                AND  A
C148 C8                RET  Z
C149 CD5702            CALL 0257H           ; display a character
C14C 23                INC  HL
C14D 18F7              JR   DSPMSG
                ;
                ; message area
                ;
C14F 0D0A0A69 PRMPT1:DEFB 0DH,0AH,0AH,'integer of x% ? ',0
C153 6E746567
C157 6572206F
C15B 66207825
C15F 203F2000
C163 696E7465 PRMPT2:DEFB 'integer of y% ? ',0
C167 67657220
C16B 6F662079
C16F 25203F20
C173 00
C174 78252065 ANS1:  DEFB 'x% equal y%',0
C178 7175616C
C17C 20792500
C180 78252067 ANS2:  DEFB 'x% greater than y%',0
C184 72656174
C188 65722074
C18C 68616E20
C190 792500
C193 7825206C ANS3:  DEFB 'x% less than y%',0
C197 65737320
C19B 7468616E
C19F 20792500
                ;
C1A3                   END


*GC100


integer of x% ? 5
integer of y% ? -5
x% greater than y%

integer of x% ? 5000
integer of y% ? 5000.4
x% equal y%

integer of x% ? 5000
integer of y% ? 5000.5
x% equal y%

integer of x% ? &H7FFF
integer of y% ? &H8000
x% greater than y%

integer of x% ? 40000
Overflow
Ok
```

２７７８Ｈ　倍精度型実数の比較

アドレス	２７７８Ｈ
機　　能	倍精度型実数の比較
レジスタ	ＡＦ－－ＤＥ－－

解　　説　フローティング・アキュムレータ（Ｆ０Ａ４Ｈ～Ｆ０ＡＢＨ番地）に格納された倍精度型の実数を第１パラメータとし，サブ・フローティング・アキュムレータ（Ｆ０ＡＥＨ～Ｆ０Ｂ５Ｈ番地）に格納された倍精度型の実数を第２パラメータとして，両者の比較結果をアキュムレータに格納し，ゼロ・フラグ（Ｚ）およびキャリ・フラグ（ＣＹ）に影響を与えてもどります．

比較結果とアキュムレータに与える数値の関係は次のとおりです．

第１パラメータ＜第２パラメータの場合，比較結果に０１Ｈを与えます．

第１パラメータ＝第２パラメータの場合，比較結果に００Ｈを与えます．

第１パラメータ＞第２パラメータの場合，比較結果にＦＦＨを与えます．

サンプル

```
                ;
                ; --- compare for double precision ---
                ;
                        ORG   0C100H
                ;
C100 2160C1     START:  LD    HL,PRMPT1
C103 CDED52             CALL  52EDH               ; display message
C106 CD7E1B             CALL  1B7EH               ; screen editor
C109 DA665C             JP    C,5C66H             ; system monitor if stop key
C10C 23                 INC   HL
C10D CDBE2B             CALL  2BBEH               ; convert into binary code
C110 CDDF27             CALL  27DFH               ; convert into double precision
                ;
C113 21A4F0             LD    HL,0F0A4H           ;
C116 11C9C1             LD    DE,BUFFER           ; store first argument into buffer
C119 010800             LD    BC,8                ;
C11C EDB0               LDIR
                ;
C11E 217DC1             LD    HL,PRMPT2
C121 CDED52             CALL  52EDH               ; display message
C124 CD7E1B             CALL  1B7EH               ; screen editor
C127 DA665C             JP    C,5C66H             ; system monitor if stop key
C12A 23                 INC   HL
C12B CDBE2B             CALL  2BBEH               ; convert into binary code
C12E CDDF27             CALL  27DFH               ; convert into double precision
                ;
C131 21A4F0             LD    HL,0F0A4H           ;
C134 11AEF0             LD    DE,0F0AEH           ; set second argument into sub-facc
C137 010800             LD    BC,8                ;
C13A EDB0               LDIR
                ;
C13C 21C9C1             LD    HL,BUFFER           ;
C13F 11A4F0             LD    DE,0F0A4H           ; set first argument into facc
C142 010800             LD    BC,8                ;
C145 EDB0               LDIR
                ;
C147 CD7827             CALL  2778H               ;*compare for double precision
```

```
                ;
C14A 2005               JR    NZ,NEQUAL
C14C 2197C1             LD    HL,ANS1          ; x# equal y#
C14F 180A               JR    DSPLAY
                ;
C151 3005       NEQUAL:JR     NC,LESS
C153 21A4C1             LD    HL,ANS2          ; x# greater than y#
C156 1803               JR    DSPLAY
                ;
C158 21B8C1     LESS:   LD    HL,ANS3          ; x# less than y#
                ;
C15B CDED52     DSPLAY:CALL 52EDH              ; display answer
                ;
C15E 18A0               JR    START
                ;
                ; message area
                ;
C160 0D0A0A64 PRMPT1:DEFB 0DH,0AH,0AH,'double precision of x# ? ',0
C164 6F75626C
C168 65207072
C16C 65636973
C170 696F6E20
C174 6F662078
C178 23203F20
C17C 00
C17D 646F7562 PRMPT2:DEFB 'double precision of y# ? ',0
C181 6C652070
C185 72656369
C189 73696F6E
C18D 206F6620
C191 7923203F
C195 2000
C197 78232065 ANS1:  DEFB 'x# equal y# ',0
C19B 7175616C
C19F 20792320
C1A3 00
C1A4 78232067 ANS2:  DEFB 'x# greater than y# ',0
C1A8 72656174
C1AC 65722074
C1B0 68616E20
C1B4 79232000
C1B8 7823206C ANS3:  DEFB 'x# less than y# ',0
C1BC 65737320
C1C0 7468616E
C1C4 20792320
C1C8 00
                ;
                ; buffer for the first argument
                ;
C1C9            BUFFER:END


*GC100


double precision of x# ? 1.11111111111111
double precision of y# ? 1.111111111111111
x# less than y#

double precision of x# ? &h7fff
double precision of y# ? &h8000
x# greater than y#

double precision of x# ?

*
```

277FH　整数型への変換

アドレス	277FH
機　　能	整数型への変換
レジスタ	AFBCDEHL

解　　説　フローティング・アキュムレータに格納された数値データを，整数型に変換し，レジスタHLにも格納してもどります.

フローティング・アキュムレータがすでに整数型であった場合には何もせずにもどりますが，文字型（EF45H番地の内容が03H）であった場合には“Type mismatch”エラーが発生し，また数値データが整数型として表現できる範囲内（－32768～32767）を越える場合には“Overflow”エラーが発生するため，インタプリタの制御下に入ります.

サンプル

```
                ;
                ; --- convert into integer ---
                ;
                       ORG   0C100H
                ;
C100 212CC1     START: LD    HL,PROMPT
C103 CD23C1            CALL  DSPMSG
C106 CD7E1B            CALL  1B7EH              ; screen editor
C109 DA665C            JP    C,5C66H            ; jump to system monitor if stop key
C10C 23                INC   HL
C10D CDBE2B            CALL  2BBEH              ; convert into binary code
                ;
C110 CD7F27            CALL  277FH              ;*convert facc into integer
C113 CD222D            CALL  2D22H              ; convert facc into character code
                ;
C116 E5                PUSH  HL
C117 213BC1            LD    HL,ANSWER
C11A CD23C1            CALL  DSPMSG
C11D E1                POP   HL
                ;
C11E CD23C1            CALL  DSPMSG             ; display result
C121 18DD              JR    START
                ;
                ; display message subroutine
                ;
C123 7E         DSPMSG:LD    A,(HL)
C124 A7                AND   A
C125 C8                RET   Z
C126 CD5702            CALL  0257H              ; display a character
C129 23                INC   HL
C12A 18F7              JR    DSPMSG
                ;
                ; message area
                ;
C12C 0D0A0A61   PROMPT:DEFB 0DH,0AH,0AH,'a number ? ',0
C130 206E756D
C134 62657220
C138 3F2000
C13B 616E2069   ANSWER:DEFB 'an integer : ',0
C13F 6E746567
C143 6572203A
C147 2000
```

```
C149             ;          END
```

```
*GC100

a number ? 5
an integer :  5

a number ? 5.4
an integer :  5

a number ? 5.5
an integer :  5

a number ? -5.4
an integer : -6

a number ? -5.5
an integer : -6

a number ? 40000
Overflow
Ok
```

２７８ＣＨ　単精度実数型から整数型への変換

アドレス	２７８ＣＨ
機　能	単精度実数型から整数型への変換
レジスタ	ＡＦＢＣＤＥＨＬ

解　説　フローティング・アキュムレータに格納された単精度実数型の数値データを，整数型に変換し，レジスタＨＬにも格納してもどります.

ただし，フローティング・アキュムレータに格納された数値データが整数型として表現できる範囲内(－３２７６８～３２７６７）を越える場合には“Overflow”エラーが発生し，インタプリタの制御下に入ります.

サンプル

```
                ;
                ; --- convert single precision number into integer ---
                ;
                        ORG   0C100H
                ;
C100 213BC1     START:  LD    HL,PROMPT
C103 CD32C1             CALL  DSPMSG
C106 CD7E1B             CALL  1B7EH             ; screen editor
C109 DA665C             JP    C,5C66H           ; jump to system monitor if stop key
C10C 23                 INC   HL
C10D CDBE2B             CALL  2BBEH             ; convert into binary code
                ;
C110 3A45EF             LD    A,(0EF45H)        ;
C113 FE04               CP    04H               ; error if not single precision
C115 2013               JR    NZ,ERROR          ;
                ;
C117 CD8C27             CALL  278CH             ;*single precision into integer
C11A CD222D             CALL  2D22H             ; convert facc into character code
                ;
C11D E5                 PUSH  HL
C11E 2159C1             LD    HL,ANSWER
C121 CD32C1             CALL  DSPMSG
C124 E1                 POP   HL
                ;
C125 CD32C1             CALL  DSPMSG            ; display an integer
C128 18D6               JR    START
                ;
C12A 2164C1     ERROR:  LD    HL,MSGERR
C12D CD32C1             CALL  DSPMSG
C130 18CE               JR    START
                ;
                ; display message subroutine
                ;
                ;    destroys : register a,f,h,l
                ;
C132 7E         DSPMSG:LD     A,(HL)
C133 A7                 AND   A
C134 C8                 RET   Z
C135 CD5702             CALL  0257H             ; display a character
C138 23                 INC   HL
C139 18F7               JR    DSPMSG
                ;
                ; message area
                ;
C13B 0D0A0A73   PROMPT:DEFB 0DH,0AH,0AH,'single precision number ? ',0
C13F 696E676C
```

```
C143 65207072
C147 65636973
C14B 696F6E20
C14F 6E756D62
C153 6572203F
C157 2000
C159 696E7465 ANSWER:DEFB 'integer : ',0
C15D 67657220
C161 3A2000
C164 6E6F7420 MSGERR:DEFB 'not single precision !!',07H,0
C168 73696E67
C16C 6C652070
C170 72656369
C174 73696F6E
C178 20212107
C17C 00
                   ;
C17D                     END


*GC100


single precision number ? 3
not single precision !!

single precision number ? 3!
integer :  3

single precision number ? 3#
not single precision !!

single precision number ? -5000
not single precision !!

single precision number ? -5000!
integer : -5000

single precision number ? -5000#
not single precision !!

single precision number ? 50000
Overflow
Ok
```

２７９ＣＨ　整数の格納

アドレス	２７９ＣＨ
機　　能	整数の格納
レジスタ	Ａーーーーーー

解　　説　レジスタＨＬに格納された有符号１６ビットの整数を，整数型のフローティング・アキュムレータ（Ｆ０Ａ８Ｈ～Ｆ０Ａ９Ｈ番地）に格納し，ＥＦ４５Ｈ番地にも整数型を示す０２Ｈを入れてもどります．

サンプル

```
                ;
                ; --- numbers with a byte ---
                ;
                        ORG   0C100H
                ;
C100 3E00               LD    A,0
                ;
C102 F5         LOOP:   PUSH  AF
                ;
C103 6F                 LD    L,A
C104 CD2CC1             CALL  DSPACC
C107 2600               LD    H,0
C109 CD9C27             CALL  279CH           ;*store register hl into facc
                ;
C10C 3E80               LD    A,80H           ;
C10E 0605               LD    B,5             ; appoint format
C110 0E00               LD    C,0             ;
                ;
C112 CD232D             CALL  2D23H           ; convert into character code
C115 CD43C1             CALL  DSPMSG          ; display the integer decimal
                ;
C118 3E20               LD    A,' '
C11A CD5702             CALL  0257H
C11D CD5702             CALL  0257H
C120 CD5702             CALL  0257H
                ;
C123 F1                 POP   AF
                ;
C124 3C                 INC   A
C125 FE00               CP    00H
C127 20D9               JR    NZ,LOOP
                ;
C129 C3665C             JP    5C66H           ; jump to system monitor
                ;
                ; display accumulator hexa decimal subroutine
                ;
                ;   destroys : register a,f,b
                ;
C12C 47         DSPACC:LD     B,A
C12D 0F                 RRCA
C12E 0F                 RRCA
C12F 0F                 RRCA
C130 0F                 RRCA
C131 CD35C1             CALL  HALF
C134 78                 LD    A,B
                ;
C135 E60F       HALF:   AND   0FH
C137 FE0A               CP    10
C139 3802               JR    C,NUMBER
C13B C607               ADD   A,7
```

```
                    ;
C13D C630           NUMBER:ADD   A,30H
C13F CD5702                CALL  0257H             ; display a character
C142 C9                    RET
                    ;
                    ; display message subroutine
                    ;
                    ;    destroys : register a,f,h,l
                    ;
C143 7E             DSPMSG:LD    A,(HL)
C144 A7                    AND   A
C145 C8                    RET   Z
C146 CD5702                CALL  0257H             ; display a character
C149 23                    INC   HL
C14A 18F7                  JR    DSPMSG
                    ;
C14C                       END
```

```
*GC100
00    0  01    1  02    2  03    3  04    4  05    5  06    6  07    7
08    8  09    9  0A   10  0B   11  0C   12  0D   13  0E   14  0F   15
10   16  11   17  12   18  13   19  14   20  15   21  16   22  17   23
18   24  19   25  1A   26  1B   27  1C   28  1D   29  1E   30  1F   31
20   32  21   33  22   34  23   35  24   36  25   37  26   38  27   39
28   40  29   41  2A   42  2B   43  2C   44  2D   45  2E   46  2F   47
30   48  31   49  32   50  33   51  34   52  35   53  36   54  37   55
38   56  39   57  3A   58  3B   59  3C   60  3D   61  3E   62  3F   63
40   64  41   65  42   66  43   67  44   68  45   69  46   70  47   71
48   72  49   73  4A   74  4B   75  4C   76  4D   77  4E   78  4F   79
50   80  51   81  52   82  53   83  54   84  55   85  56   86  57   87
58   88  59   89  5A   90  5B   91  5C   92  5D   93  5E   94  5F   95
60   96  61   97  62   98  63   99  64  100  65  101  66  102  67  103
68  104  69  105  6A  106  6B  107  6C  108  6D  109  6E  110  6F  111
70  112  71  113  72  114  73  115  74  116  75  117  76  118  77  119
78  120  79  121  7A  122  7B  123  7C  124  7D  125  7E  126  7F  127
80  128  81  129  82  130  83  131  84  132  85  133  86  134  87  135
88  136  89  137  8A  138  8B  139  8C  140  8D  141  8E  142  8F  143
90  144  91  145  92  146  93  147  94  148  95  149  96  150  97  151
98  152  99  153  9A  154  9B  155  9C  156  9D  157  9E  158  9F  159
A0  160  A1  161  A2  162  A3  163  A4  164  A5  165  A6  166  A7  167
A8  168  A9  169  AA  170  AB  171  AC  172  AD  173  AE  174  AF  175
B0  176  B1  177  B2  178  B3  179  B4  180  B5  181  B6  182  B7  183
B8  184  B9  185  BA  186  BB  187  BC  188  BD  189  BE  190  BF  191
C0  192  C1  193  C2  194  C3  195  C4  196  C5  197  C6  198  C7  199
C8  200  C9  201  CA  202  CB  203  CC  204  CD  205  CE  206  CF  207
D0  208  D1  209  D2  210  D3  211  D4  212  D5  213  D6  214  D7  215
D8  216  D9  217  DA  218  DB  219  DC  220  DD  221  DE  222  DF  223
E0  224  E1  225  E2  226  E3  227  E4  228  E5  229  E6  230  E7  231
E8  232  E9  233  EA  234  EB  235  EC  236  ED  237  EE  238  EF  239
F0  240  F1  241  F2  242  F3  243  F4  244  F5  245  F6  246  F7  247
F8  248  F9  249  FA  250  FB  251  FC  252  FD  253  FE  254  FF  255
*
```

27A1H　フローティング・アキュムレータの型指定

アドレス	27A1H
機　能	フローティング・アキュムレータの型指定
レジスタ	―――――――――

解　説　アキュムレータに格納された数値を，フローティング・アキュムレータの型としてＥＦ４５Ｈ番地に格納します．

ＥＡ４５Ｈ番地に格納する数値と，フローティング・アキュムレータの型の関係はつぎのとおりです．

　ＥＡ４５Ｈ番地に与える数値が０２Ｈの場合，整数型を示します．
　ＥＡ４５Ｈ番地に与える数値が０３Ｈの場合，文字型を示します．
　ＥＡ４５Ｈ番地に与える数値が０４Ｈの場合，単精度実数型を示します．
　ＥＡ４５Ｈ番地に与える数値が０８Ｈの場合，倍精度実数型を示します．

サンプル

```
                ;
                ; --- convert intermediate code into the number ---
                ;
                        ORG   0C100H
                ;
C100 1171C1     START:  LD    DE,PRMPT1
C103 CD68C1             CALL  DSPMSG
                ;
C106 CD750F             CALL  0F75H           ; input a character from keyboard
C109 CD5702             CALL  0257H           ; echo-back a character
C10C FE03               CP    03H
C10E CA665C             JP    Z,5C66H         ; jump to system monitor if stop key
C111 D630               SUB   '0'             ; convert into binary code
C113 CDA127             CALL  27A1H           ;*set type of floating accumulator
C116 47                 LD    B,A             ; set loop counter for input
                ;
C117 21A8F0             LD    HL,0F0A8H       ; top address for single precision
                ;
C11A FE08               CP    8
C11C 2003               JR    NZ,NEXTIN
C11E 21A4F0             LD    HL,0F0A4H       ; top address for double precision
                ;
C121 118AC1     NEXTIN:LD     DE,PRMPT2
C124 CD68C1             CALL  DSPMSG
                ;
C127 CD750F             CALL  0F75H           ; input a character from keyboard
C12A CD5702             CALL  0257H           ; echo-back a character
C12D CD5CC1             CALL  HEXBIN          ; convert hexa code into binary
C130 4F                 LD    C,A
                ;
C131 CD750F             CALL  0F75H           ; input a character from keyboard
C134 CD5702             CALL  0257H           ; echo-back a character
C137 FE0D               CP    0DH
C139 280D               JR    Z,CRCODE
C13B CD5CC1             CALL  HEXBIN          ; convert hexa code into binary
                ;
C13E CB01               RLC   C
C140 CB01               RLC   C
C142 CB01               RLC   C
C144 CB01               RLC   C
```

```
C146 81                 ADD  A,C
C147 4F                 LD   C,A
                ;
C148 71         CRCODE:LD   (HL),C
C149 23                 INC  HL
C14A 10D5               DJNZ NEXTIN
                ;
C14C CD222D             CALL 2D22H          ; convert into character code
C14F E5                 PUSH HL             ; push top address onto stack
                ;
C150 11A2C1             LD   DE,ANSWER
C153 CD68C1             CALL DSPMSG
                ;
C156 D1                 POP  DE             ; pop top address from stack
C157 CD68C1             CALL DSPMSG         ; display the number
                ;
C15A 18A4               JR   START
                ;
                ; convert hexa code into binary code subroutine
                ;
                ;    inputs   : character code in accumulator
                ;    outputs  : binary code in accumulator
                ;    destroys : register a,f
                ;
C15C FE3A       HEXBIN:CP    '9'+1
C15E 3805               JR   C,HEXBI1
C160 E60F               AND  0FH
C162 C609               ADD  A,9
C164 C9                 RET
                ;
C165 E60F       HEXBI1:AND  0FH
C167 C9                 RET
                ;
                ; display message subroutine
                ;
                ;    destroys : register a,f,d,e
                ;
C168 1A         DSPMSG:LD   A,(DE)
C169 A7                 AND  A
C16A C8                 RET  Z
C16B CD5702             CALL 0257H
C16E 13                 INC  DE
C16F 18F7               JR   DSPMSG
                ;
                ; message area
                ;
C171 0D0A0A74 PRMPT1:DEFB 0DH,0AH,0AH,'type of the number ? ',0
C175 79706520
C179 6F662074
C17D 6865206E
C181 756D6265
C185 72203F20
C189 00
C18A 0D0A2020 PRMPT2:DEFB 0DH,0AH,'          on memory ? ',0
C18E 20202020
C192 2020206F
C196 6E206D65
C19A 6D6F7279
C19E 203F2000
C1A2 0D0A2020 ANSWER:DEFB 0DH,0AH,'         the number : ',0
C1A6 20202020
C1AA 20207468
C1AE 65206E75
C1B2 6D626572
C1B6 203A2000
                ;
C1BA                    END
```

```
*GC100

 type of the number ? 2
           on memory ? 00
           on memory ? 00
          the number :  0

 type of the number ? 2
           on memory ? 01
           on memory ? 00
          the number :  1

 type of the number ? 2
           on memory ? 50
           on memory ? 00
          the number :  80

 type of the number ? 2
           on memory ? ff
           on memory ? ff
          the number : -1

 type of the number ? 4
           on memory ? 11
           on memory ? 11
           on memory ? 11
           on memory ? 11
          the number :  2.18272E-34

 type of the number ? 8
           on memory ? 00
           on memory ? 00
           on memory ? 00
           on memory ? 00
           on memory ? 00
           on memory ? 00
           on memory ? 00
           on memory ? 84
          the number :  8

 type of the number ? 2
           on memory ? 00
           on memory ? 80
          the number : -32768

 type of the number ?
*
```

ＦＡＣＣの構成

ＢＡＳＩＣインタプリタは，浮動小数点の演算機能が用意されていないμＰＤ７８０（Ｚ８０コンパチブル）で浮動小数点の演算を実行するため，フローティング・アキュムレータ（ＦＡＣＣ＝ Floating Point Accumulator）と呼ぶ８バイトの仮想レジスタをメモリ上に用意し，すべてソフトウェアによって浮動小数点の演算処理を実現しています．

このフローティング・アキュムレータは，Ｆ０Ａ４Ｈ～Ｆ０ＡＢＨ番地に置かれ，ＥＦ４５Ｈ番地にはデータの型を表す数値が与えられます．

●整数型

番地	内容
ＥＦ４５Ｈ番地	０２Ｈ
Ｆ０Ａ８Ｈ番地	下位８ビット
Ｆ０Ａ９Ｈ番地	上位８ビット

●単精度実数型

番地	内容
ＥＦ４５Ｈ番地	０４Ｈ
Ｆ０Ａ８Ｈ番地	仮数部下位８ビット
Ｆ０Ａ９Ｈ番地	仮数部中位８ビット
Ｆ０ＡＡＨ番地	仮数部上位８ビット
Ｆ０ＡＢＨ番地	指数部８ビット

●倍精度実数型

番地	内容
ＥＦ４５Ｈ番地	０８Ｈ
Ｆ０Ａ４Ｈ番地	仮数部最下位８ビット
Ｆ０Ａ５Ｈ番地	¦
Ｆ０Ａ６Ｈ番地	¦
Ｆ０Ａ７Ｈ番地	¦
Ｆ０Ａ８Ｈ番地	¦
Ｆ０Ａ９Ｈ番地	¦
Ｆ０ＡＡＨ番地	仮数部最上位８ビット
Ｆ０ＡＢＨ番地	指数部８ビット

２７Ｂ３Ｈ　単精度実数型への変換

アドレス	２７Ｂ３Ｈ
機　能	単精度実数型への変換
レジスタ	ＡＦＢＣＤＥＨＬ

解　説　フローティング・アキュムレータに格納された数値データを，単精度実数型に変換してもどります．

フローティング・アキュムレータがすでに単精度実数型であった場合には何もせずにもどりますが，文字型（ＥＦ４５Ｈ番地の内容が０３Ｈ）であった場合には，“Type mismatch” エラーが発生し，インタプリタの制御下に入ります．

サンプル

```
               ;
               ; --- convert into single precision ---
               ;
                      ORG   0C100H
               ;
C100 212CC1    START: LD    HL,PROMPT
C103 CD23C1           CALL  DSPMSG
C106 CD7E1B           CALL  1B7EH           ; screen editor
C109 DA665C           JP    C,5C66H         ; jump to system monitor if stop key
C10C 23               INC   HL
C10D CDBE2B           CALL  2BBEH           ; convert into binary code
               ;
C110 CDB327           CALL  27B3H           ;*convert facc into single precision
C113 CD222D           CALL  2D22H           ; convert facc into character code
               ;
C116 E5               PUSH  HL
C117 213BC1           LD    HL,ANSWER
C11A CD23C1           CALL  DSPMSG
C11D E1               POP   HL
               ;
C11E CD23C1           CALL  DSPMSG          ; display single precision number
C121 18DD             JR    START
               ;
               ; display message subroutine
               ;
               ;    destroys : register a,f,h,l
               ;
C123 7E        DSPMSG:LD    A,(HL)
C124 A7               AND   A
C125 C8               RET   Z
C126 CD5702           CALL  0257H           ; display a character
C129 23               INC   HL
C12A 18F7             JR    DSPMSG
               ;
               ; message area
               ;
C12C 0D0A0A61  PROMPT:DEFB 0DH,0AH,0AH,'a number ? ',0
C130 206E756D
C134 62657220
C138 3F2000
C13B 61207369  ANSWER:DEFB 'a single precision number : ',0
C13F 6E676C65
C143 20707265
C147 63697369
C14B 6F6E206E
C14F 756D6265
C153 72203A20
```

```
C157 00
                  ;
C158                     END


*GC100

a number ? 1
a single precision number :  1

a number ? 1.00001
a single precision number :  1.00001

a number ? 1.0000000001
a single precision number :  1

a number ? 3333333333333333333333333333
a single precision number :  3.33333E+27

a number ? 0
a single precision number :  0

a number ? 0.4
a single precision number :  .4

a number ?

*
```

27BDH 倍精度実数型から単精度実数型への変換

アドレス	27BDH
機　能	倍精度実数型から単精度実数型への変換
レジスタ	AFBCDEHL

解　説　フローティング・アキュムレータに格納された倍精度実数型の数値データを，単精度実数型に変換してもどります．

サンプル

```
                ;
                ; --- convert double precision into single precision ---
                ;
                        ORG   0C100H
                ;
C100 213BC1     START: LD    HL,PROMPT
C103 CD32C1            CALL  DSPMSG
C106 CD7E1B            CALL  1B7EH            ; screen editor
C109 DA665C            JP    C,5C66H          ; jump to system monitor if stop key
C10C 23                INC   HL
C10D CDBE2B            CALL  2BBEH            ; convert into binary code
                ;
C110 3A45EF            LD    A,(0EF45H)       ;
C113 FE08              CP    08H              ; error if not double precision
C115 2013              JR    NZ,ERROR         ;
                ;
C117 CDBD27            CALL  27BDH            ;*double into single precision
C11A CD222D            CALL  2D22H            ; convert facc into character code
                ;
C11D E5                PUSH  HL
C11E 2159C1            LD    HL,ANSWER
C121 CD32C1            CALL  DSPMSG
C124 E1                POP   HL
                ;
C125 CD32C1            CALL  DSPMSG           ; display single precision number
C128 18D6              JR    START
                ;
C12A 2174C1     ERROR: LD    HL,MSGERR
C12D CD32C1            CALL  DSPMSG
C130 18CE              JR    START
                ;
                ; display message subroutine
                ;
                ;     destroys : register a,f,h,l
                ;
C132 7E         DSPMSG:LD    A,(HL)
C133 A7                AND   A
C134 C8                RET   Z
C135 CD5702            CALL  0257H            ; display a character
C138 23                INC   HL
C139 18F7              JR    DSPMSG
                ;
                ; message area
                ;
C13B 0D0A0A64   PROMPT:DEFB 0DH,0AH,0AH,'double precision number ? ',0
C13F 6F75626C
C143 65207072
C147 65636973
C14B 696F6E20
C14F 6E756D62
C153 6572203F
C157 2000
C159 73696E67   ANSWER:DEFB 'single precision number : ',0
```

```
C15D 6C652070
C161 72656369
C165 73696F6E
C169 206E756D
C16D 62657220
C171 3A2000
C174 6E6F7420 MSGERR:DEFB 'not double precision !!',07H,0
C178 646F7562
C17C 6C652070
C180 72656369
C184 73696F6E
C188 20212107
C18C 00
                 ;
C18D                    END


*GC100


double precision number ? 33333
not double precision !!

double precision number ? 3333333333
single precision number :  3.33333E+09

double precision number ? 333333333333333333333333333
single precision number :  3.33333E+26

double precision number ? 5%
not double precision !!

double precision number ? 5!
not double precision !!

double precision number ? 5#
single precision number :  5

double precision number ?

*
```

27D0H　整数型から単精度実数型への変換

アドレス	27D0H
機　能	整数型から単精度実数型への変換
レジスタ	AFBCDEHL

解　説　フローティング・アキュムレータに格納された整数型の数値データを，単精度実数型に変換してもどります．

サンプル

```
               ;
               ; --- convert integer into single precision ---
               ;
                      ORG   0C100H
               ;
C100 213BC1    START: LD    HL,PROMPT
C103 CD32C1           CALL  DSPMSG
C106 CD7E1B           CALL  1B7EH              ; screen editor
C109 DA665C           JP    C,5C66H            ; jump to system monitor if stop key
C10C 23               INC   HL
C10D CDBE2B           CALL  2BBEH              ; convert into binary code
               ;
C110 3A45EF           LD    A,(0EF45H)         ;
C113 FE02             CP    02H                ; error if not integer
C115 2013             JR    NZ,ERROR           ;
               ;
C117 CDD027           CALL  27D0H              ;*integer into single precision
C11A CD222D           CALL  2D22H              ; convert facc into character code
               ;
C11D E5               PUSH  HL
C11E 2149C1           LD    HL,ANSWER
C121 CD32C1           CALL  DSPMSG
C124 E1               POP   HL
               ;
C125 CD32C1           CALL  DSPMSG`            ; display single precision number
C128 18D6             JR    START
               ;
C12A 2164C1    ERROR: LD    HL,MSGERR
C12D CD32C1           CALL  DSPMSG
C130 18CE             JR    START
               ;
               ; display message subroutine
               ;
               ;    destroys : register a,f,h,l
               ;
C132 7E        DSPMSG:LD    A,(HL)
C133 A7               AND   A
C134 C8               RET   Z
C135 CD5702           CALL  0257H              ; display a character
C138 23               INC   HL
C139 18F7             JR    DSPMSG
               ;
               ; message area
               ;
C13B 0D0A0A69  PROMPT:DEFB 0DH,0AH,0AH,'integer ? ',0
C13F 6E746567
C143 6572203F
C147 2000
C149 73696E67  ANSWER:DEFB 'single precision number : ',0
C14D 6C652070
C151 72656369
C155 73696F6E
C159 206E756D
```

```
C15D 62657220
C161 3A2000
C164 6E6F7420 MSGERR:DEFB 'not integer !!',07H,0
C168 696E7465
C16C 67657220
C170 21210700
                  ;
C174                      END
```

```
*GC100

integer ? 0
single precision number :  0

integer ? 1
single precision number :  1

integer ? 1#
not integer !!

integer ? 1.5
not integer !!

integer ? -5000
single precision number : -5000

integer ? &hffff
single precision number : -1

integer ?

*
```

２７ＤＦＨ　倍精度実数型への変換

アドレス	２７ＤＦＨ
機　能	倍精度実数型への変換
レジスタ	ＡＦＢＣＤＥＨＬ

解　説　フローティング・アキュムレータに格納された数値データを，倍精度実数型に変換してもどります.

フローティング・アキュムレータがすでに倍精度実数型であった場合には何もせずにもどりますが，文字型（ＥＦ４５Ｈ番地の内容が０３Ｈ）であった場合には，“Type mismatch” エラーが発生しインタプリタの制御下に入ります.

サンプル

```
                ;
                ; --- convert into double precision ---
                ;
                        ORG  0C100H
                ;
C100 212CC1     START: LD   HL,PROMPT
C103 CD23C1            CALL DSPMSG
C106 CD7E1B            CALL 1B7EH             ; screen editor
C109 DA665C            JP   C,5C66H           ; jump to system monitor if stop key
C10C 23                INC  HL
C10D CDBE2B            CALL 2BBEH             ; convert into binary code
                ;
C110 CDDF27            CALL 27DFH             ;*convert facc into double precision
C113 CD222D            CALL 2D22H             ; convert facc into character code
                ;
C116 E5                PUSH HL
C117 213BC1            LD   HL,ANSWER
C11A CD23C1            CALL DSPMSG
C11D E1                POP  HL
                ;
C11E CD23C1            CALL DSPMSG            ; display double precision number
C121 18DD              JR   START
                ;
                ; display message subroutine
                ;
                ;    destroys : register a,f,h,l
                ;
C123 7E         DSPMSG:LD   A,(HL)
C124 A7                AND  A
C125 C8                RET  Z
C126 CD5702            CALL 0257H             ; display a character
C129 23                INC  HL
C12A 18F7              JR   DSPMSG
                ;
                ; message area
                ;
C12C 0D0A0A61   PROMPT:DEFB 0DH,0AH,0AH,'a number ? ',0
C130 206E756D
C134 62657220
C138 3F2000
C13B 6120646F   ANSWER:DEFB 'a double precision number : ',0
C13F 75626C65
C143 20707265
C147 63697369
C14B 6F6E206E
C14F 756D6265
```

```
C153 72203A20
C157 00
                  ;
C158                       END


*GC100


a number ? 1
a double precision number :  1

a number ? 100000000000
a double precision number :  100000000000

a number ? 10000000000000000000000000000000
a double precision number :  1D+32

a number ? 1%
a double precision number :  1

a number ? 1!
a double precision number :  1

a number ? 1#
a double precision number :  1

a number ?

*
```

27E9H　単精度実数型から倍精度実数型への変換

アドレス	27E9H
機　能	単精度実数型から倍精度実数型への変換
レジスタ	A−BC−−HL

解　説　フローティング・アキュムレータに格納された単精度実数型の数値データを，倍精度実数型に変換してもどります．

サンプル

```
                  ;
                  ; --- convert single precision into double precision ---
                  ;
                          ORG   0C100H
                  ;
C100 213BC1       START:  LD    HL,PROMPT
C103 CD32C1               CALL  DSPMSG
C106 CD7E1B               CALL  1B7EH              ; screen editor
C109 DA665C               JP    C,5C66H            ; jump to system monitor if stop key
C10C 23                   INC   HL
C10D CDBE2B               CALL  2BBEH              ; convert into binary code
                  ;
C110 3A45EF               LD    A,(0EF45H)         ;
C113 FE04                 CP    04H                ; error if not single precision
C115 2013                 JR    NZ,ERROR           ;
                  ;
C117 CDE927               CALL  27E9H              ;*single into double precision
C11A CD222D               CALL  2D22H              ; convert facc into character code
                  ;
C11D E5                   PUSH  HL
C11E 2159C1               LD    HL,ANSWER
C121 CD32C1               CALL  DSPMSG
C124 E1                   POP   HL
                  ;
C125 CD32C1               CALL  DSPMSG             ; display double precision number
C128 18D6                 JR    START
                  ;
C12A 2174C1       ERROR:  LD    HL,MSGERR
C12D CD32C1               CALL  DSPMSG
C130 18CE                 JR    START
                  ;
                  ; display message subroutine
                  ;
                  ;     destroys : register a,f,h,l
                  ;
C132 7E           DSPMSG:LD     A,(HL)
C133 A7                   AND   A
C134 C8                   RET   Z
C135 CD5702               CALL  0257H              ; display a character
C138 23                   INC   HL
C139 18F7                 JR    DSPMSG
                  ;
                  ; message area
                  ;
C13B 0D0A0A73 PROMPT:DEFB 0DH,0AH,0AH,'single precision number ? ',0
C13F 696E676C
C143 65207072
C147 65636973
C14B 696F6E20
C14F 6E756D62
C153 6572203F
C157 2000
C159 646F7562 ANSWER:DEFB 'double precision number : ',0
```

```
C15D 6C652070
C161 72656369
C165 73696F6E
C169 206E756D
C16D 62657220
C171 3A2000
C174 6E6F7420 MSGERR:DEFB 'not single precision !!',07H,0
C178 73696E67
C17C 6C652070
C180 72656369
C184 73696F6E
C188 20212107
C18C 00
                 ;
C18D                     END


*GC100


single precision number ? 10000
not single precision !!

single precision number ? 100000
double precision number :  100000

single precision number ? 10000000000
not single precision !!

single precision number ? 123.456
double precision number :  123.4559936523438

single precision number ?

*
Ok
print cdbl(123.456)
 123.4559936523438
Ok
```

２７Ｆ２Ｈ　ＦＡＣＣの倍精度実数型指定

アドレス　２７Ｆ２Ｈ

機　　能　**フローティング・アキュムレータの倍精度実数型指定**

レジスタ　Ａ－ＢＣ－－－－

解　　説　フローティング・アキュムレータの型を格納するＥＦ４５Ｈ番地に，倍精度実数型を示す０８Ｈを入れてもどります．

具体的には，アキュムレータに０８Ｈを与えて２７Ａ１Ｈ番地にジャンプしています．

２７Ｆ５Ｈ　ＦＡＣＣの単精度実数型指定

アドレス　２７Ｆ５Ｈ

機　　能　**フローティング・アキュムレータの単精度実数型指定**

レジスタ　Ａ－－－－－－－

解　　説　フローティング・アキュムレータの型を格納するＥＦ４５Ｈ番地に，単精度実数型を示す０４Ｈを入れてもどります．

具体的には，アキュムレータに０４Ｈを与えて２７Ａ１Ｈ番地にジャンプしています．

サンプル

```
               ;
               ; --- calculate r^2*pi ---
               ;
                      ORG   0C100H
               ;
C100 214AC1    START: LD    HL,PROMPT
C103 CD41C1           CALL  DSPMSG
C106 CD7E1B           CALL  1B7EH               ; screen editor
C109 DA665C           JP    C,5C66H             ; jump to system monitor if stop key
C10C 23               INC   HL
C10D CDBE2B           CALL  2BBEH               ; convert into binary code
C110 CDB327           CALL  27B3H               ; convert into single precision
C113 CDBD26           CALL  26BDH               ; move from facc to register bcde
C116 CD4125           CALL  2541H               ; multiply r by r
C119 CDB327           CALL  27B3H               ; convert into single precision
C11C CDBD26           CALL  26BDH               ; move from facc to register bcde
               ;
C11F 21DB0F           LD    HL,0FDBH            ;
C122 22A8F0           LD    (0F0A8H),HL         ;
C125 214982           LD    HL,8249H            ;*set constant number of 3.14159
C128 22AAF0           LD    (0F0AAH),HL         ;
C12B CDF527           CALL  27F5H               ;
               ;
```

```
C12E CD4125            CALL 2541H          ; multiply r^2 by pi
C131 CD222D            CALL 2D22H          ; convert into character code
C134 E5                PUSH HL
C135 2157C1            LD   HL,ANSWER
C138 CD41C1            CALL DSPMSG
C13B E1                POP  HL
C13C CD41C1            CALL DSPMSG         ; display result
                ;
C13F 18BF              JR   START
                ;
                ; display message subroutine
                ;
                ;    destroys : register a,f,h,l
                ;
C141 7E         DSPMSG:LD   A,(HL)
C142 A7                AND  A
C143 C8                RET  Z
C144 CD5702            CALL 0257H          ; display a character
C147 23                INC  HL
C148 18F7              JR   DSPMSG
                ;
                ; message area
                ;
C14A 0D0A0A72 PROMPT:DEFB 0DH,0AH,0AH,'r       ? ',0
C14E 20202020
C152 20203F20
C156 00
C157 725E322A ANSWER:DEFB 'r^2*pi : ',0
C15B 7069203A
C15F 2000
                ;
C161                   END


*GC100


r       ? 0.001
r^2*pi :  3.14159E-06

r       ? 0.1
r^2*pi :  .0314159

r       ? 1
r^2*pi :  3.14159

r       ? 10
r^2*pi :  314.159

r       ? 100
r^2*pi :  31415.9

r       ? 1000
r^2*pi :  3.14159E+06

r       ?

*
```

２８２ＣＨ　整数部の計算

アドレス	２８２ＣＨ
機　能	整数部の計算
レジスタ	ＡＦＢＣＤＥＨＬ

解　説　フローティング・アキュムレータに格納された数値データの小数点以下を取り去った値（ＦＩＸ）を求め，同じくフローティング・アキュムレータに与えてもどります．

このシステム・サブルーチンによる演算結果と，２８３ＦＨ番地のシステム・サブルーチンによる演算結果は，フローティング・アキュムレータに格納された数値データが正のとき等しくなりますが，負となった場合前者は小数部分を取り去ってしまいます．したがって，この場合は前者で得られる値が後者で得られる値よりも大きくなります．

サンプル

```
                ;
                ; --- fix function test progtam ---
                ;
                        ORG   0C100H
                ;
C100 2137C1     START: LD     HL,PROMPT
C103 CD2EC1            CALL   DSPMSG
                ;
C106 CD7E1B            CALL   1B7EH           ; screen editor
C109 DA665C            JP     C,5C66H         ; system monitor if stop key
                ;
C10C 23                INC    HL
C10D E5                PUSH   HL
                ;
C10E CDBE2B            CALL   2BBEH           ; convert into binary code
C111 CD2C28            CALL   282CH           ;*abs function
C114 CD222D            CALL   2D22H           ; convert into character code
                ;
C117 EB                EX     DE,HL           ; top address of result in reg-de
                ;
C118 2145C1            LD     HL,ANS1
C11B CD2EC1            CALL   DSPMSG
                ;
C11E E1                POP    HL              ; top address of a input number
C11F CD2EC1            CALL   DSPMSG          ; display a input number again
                ;
C122 2151C1            LD     HL,ANS2
C125 CD2EC1            CALL   DSPMSG
                ;
C128 EB                EX     DE,HL           ; top address of result in reg-hl
C129 CD2EC1            CALL   DSPMSG          ; display result
                ;
C12C 18D2              JR     START
                ;
                ; display message subroutine
                ;
                ;    destroys : register a,f,h,l
                ;
C12E 7E         DSPMSG:LD     A,(HL)
C12F A7                AND    A
```

```
C130 C8                 RET  Z
C131 CD5702             CALL 0257H             ; display a character
C134 23                 INC  HL
C135 18F7               JR   DSPMSG
              ;
              ; message area
              ;
C137 0D0A6120 PROMPT:DEFB 0DH,0AH,'a number ? ',0
C13B 6E756D62
C13F 6572203F
C143 2000
C145 20202020 ANS1:   DEFB '     fix ( ',0
C149 20666978
C14D 20282000
C151 20292069 ANS2:   DEFB ' ) is equal to ',0
C155 73206571
C159 75616C20
C15D 746F2000
              ;
C161                    END


*GC100

a number ? -2
     fix ( -2 ) is equal to -2
a number ? -1.6
     fix ( -1.6 ) is equal to -1
a number ? -1.5
     fix ( -1.5 ) is equal to -1
a number ? -1.4
     fix ( -1.4 ) is equal to -1
a number ? 0
     fix ( 0 ) is equal to  0
a number ? 1.4
     fix ( 1.4 ) is equal to  1
a number ? 1.5
     fix ( 1.5 ) is equal to  1
a number ? 1.6
     fix ( 1.6 ) is equal to  1
a number ? 2
     fix ( 2 ) is equal to  2
a number ?

*
```

２８３ＦＨ　小数点以下を切り捨てた整数値の計算

アドレス	２８３ＦＨ
機　能	小数点以下を切り捨てた整数値の計算
レジスタ	ＡＦＢＣＤＥＨＬ

解　説　フローティング・アキュムレータに格納された数値データの値を超えない最大の整数（ＩＮＴ）をもとめ，同じくフローティング・アキュムレータに与えてもどります．

サンプル

```
                ;
                ; --- int function test progtam ---
                ;
                        ORG  0C100H
                ;
C100 2137C1     START:  LD   HL,PROMPT
C103 CD2EC1             CALL DSPMSG
                ;
C106 CD7E1B             CALL 1B7EH              ; screen editor
C109 DA665C             JP   C,5C66H            ; system monitor if stop key
                ;
C10C 23                 INC  HL
C10D E5                 PUSH HL
                ;
C10E CDBE2B             CALL 2BBEH              ; convert into binary code
C111 CD3F28             CALL 283FH              ;*int function
C114 CD222D             CALL 2D22H              ; convert into character code
                ;
C117 EB                 EX   DE,HL              ; top address of result in reg-de
                ;
C118 2145C1             LD   HL,ANS1
C11B CD2EC1             CALL DSPMSG
                ;
C11E E1                 POP  HL                 ; top address of a input number
C11F CD2EC1             CALL DSPMSG             ; display a input number again
                ;
C122 2151C1             LD   HL,ANS2
C125 CD2EC1             CALL DSPMSG
                ;
C128 EB                 EX   DE,HL              ; top address of result in reg-hl
C129 CD2EC1             CALL DSPMSG             ; display result
                ;
C12C 18D2               JR   START
                ;
                ; display message subroutine
                ;
                ;    destroys : register a,f,h,l
                ;
C12E 7E         DSPMSG:LD    A,(HL)
C12F A7                 AND  A
C130 C8                 RET  Z
C131 CD5702             CALL 0257H              ; display a character
C134 23                 INC  HL
C135 18F7               JR   DSPMSG
                ;
                ; message area
                ;
C137 0D0A6120   PROMPT:DEFB 0DH,0AH,'a number ? ',0
C13B 6E756D62
C13F 6572203F
C143 2000
```

```
C145 20202020 ANS1:  DEFB '      int ( ',0
C149 20696E74
C14D 20282000
C151 20292069 ANS2:  DEFB ' ) is equal to ',0
C155 73206571
C159 75616C20
C15D 746F2000
                 ;
C161                 END


*GC100

a number ? -2
     int ( -2 ) is equal to -2
a number ? -1.6
     int ( -1.6 ) is equal to -2
a number ? -1.5
     int ( -1.5 ) is equal to -2
a number ? -1.4
     int ( -1.4 ) is equal to -2
a number ? 0
     int ( 0 ) is equal to  0
a number ? 1.4
     int ( 1.4 ) is equal to  1
a number ? 1.5
     int ( 1.5 ) is equal to  1
a number ? 1.6
     int ( 1.6 ) is equal to  1
a number ? 2
     int ( 2 ) is equal to  2
a number ?

*
```

２８Ｄ２Ｈ　整数の減算

アドレス	２８Ｄ２Ｈ
機　能	整数の減算
レジスタ	ＡＦＢＣＤＥＨＬ

解　説　レジスタＤＥに格納された有符号１６ビットの整数を被減数とし，レジスタＨＬに格納された有符号１６ビットの整数を減数として，両者の間で減算を行い，演算結果をレジスタＨＬおよびフローティング・アキュムレータ（Ｆ０Ａ８Ｈ～Ｆ０Ａ９Ｈ番地）に格納してもどります．

サンプル

```
                   ;
                   ; --- subtraction for integer ---
                   ;
                        ORG  0C100H
                   ;
C100 2148C1        START: LD   HL,PRMPT1
C103 CD3FC1             CALL DSPMSG
C106 CD7E1B             CALL 1B7EH          ; screen editor
C109 DA665C             JP   C,5C66H        ; system monitor if stop key
C10C 23                 INC  HL
C10D CDBE2B             CALL 2BBEH          ; convert into binary code
C110 CD7F27             CALL 277FH          ; convert into integer
C113 ED5BA8F0           LD   DE,(0F0A8H)    ; store 1st argument into reg-de
C117 D5                 PUSH DE
                   ;
C118 2158C1             LD   HL,PRMPT2
C11B CD3FC1             CALL DSPMSG
C11E CD7E1B             CALL 1B7EH          ; screen editor
C121 23                 INC  HL
C122 CDBE2B             CALL 2BBEH          ; convert into binary code
C125 CD7F27             CALL 277FH          ; convert into integer
C128 2AA8F0             LD   HL,(0F0A8H)    ; set 2nd argument into reg-hl
                   ;
C12B D1                 POP  DE             ; set 1st argument into reg-de
                   ;
C12C CDD228             CALL 28D2H          ;*subtraction for integer
                   ;
C12F CD222D             CALL 2D22H          ; convert into character code
C132 E5                 PUSH HL
C133 2165C1             LD   HL,ANSWER
C136 CD3FC1             CALL DSPMSG
C139 E1                 POP  HL
C13A CD3FC1             CALL DSPMSG         ; display result
                   ;
C13D 18C1               JR   START
                   ;
                   ; display message subroutine
                   ;
C13F 7E            DSPMSG:LD   A,(HL)
C140 A7                 AND  A
C141 C8                 RET  Z
C142 CD5702             CALL 0257H          ; display a character
C145 23                 INC  HL
C146 18F7               JR   DSPMSG
                   ;
                   ; message area
                   ;
C148 0D0A0A20      PRMPT1:DEFB 0DH,0AH,0AH,'      x%   ? ',0
C14C 20202020
```

```
C150 78252020
C154 203F2000
C158 20202020 PRMPT2:DEFB '      y%   ? ',0
C15C 20792520
C160 20203F20
C164 00
C165 7825202D ANSWER:DEFB 'x% - y% --> ',0
C169 20792520
C16D 2D2D3E20
C171 00
                 ;
C172                    END


*GC100


     x%   ? 123
     y%   ? 23
x% - y% -->  100

     x%   ? 123
     y%   ? 0.4
x% - y% -->  123

     x%   ? 123
     y%   ? 0.4
x% - y% -->  123

     x%   ? 123
     y%   ? 0.5
x% - y% -->  123

     x%   ?

*
```

２８ＤＤＨ　整数の加算

アドレス	２８ＤＤＨ
機　能	整数の加算
レジスタ	ＡＦＢＣＤＥＨＬ

解　説　レジスタＤＥに格納された有符号１６ビットの整数を被加数とし，レジスタＨＬに格納された有符号１６ビットの整数を加数として，両者の間で加算を行い，演算結果をレジスタＨＬおよびフローティング・アキュムレータ（Ｆ０Ａ８Ｈ～Ｆ０Ａ９Ｈ番地）に格納してもどります．

演算結果が整数型として扱い得る範囲を越えた場合には，単精度の値としてフローティング・アキュムレータに返されますが，その場合にはレジスタＨＬに与えられたパラメータは意味を持ちません．

サンプル

```
              ;
              ; --- addition for integer ---
              ;
                     ORG  0C100H
              ;
C100 2148C1   START: LD   HL,PRMPT1
C103 CD3FC1          CALL DSPMSG
C106 CD7E1B          CALL 1B7EH            ; screen editor
C109 DA665C          JP   C,5C66H          ; system monitor if stop key
C10C 23              INC  HL
C10D CDBE2B          CALL 2BBEH            ; convert into binary code
C110 CD7F27          CALL 277FH            ; convert into integer
C113 ED5BA8F0        LD   DE,(0F0A8H)      ; store 1st argument into reg-de
C117 D5              PUSH DE
              ;
C118 2158C1          LD   HL,PRMPT2
C11B CD3FC1          CALL DSPMSG
C11E CD7E1B          CALL 1B7EH            ; screen editor
C121 23              INC  HL
C122 CDBE2B          CALL 2BBEH            ; convert into binary code
C125 CD7F27          CALL 277FH            ; convert into integer
C128 2AA8F0          LD   HL,(0F0A8H)      ; set 2nd argument into reg-hl
              ;
C12B D1              POP  DE               ; set 1st argument into reg-de
              ;
C12C CDDD28          CALL 28DDH            ;*addition for integer
              ;
C12F CD222D          CALL 2D22H            ; convert into character code
C132 E5              PUSH HL
C133 2165C1          LD   HL,ANSWER
C136 CD3FC1          CALL DSPMSG
C139 E1              POP  HL
C13A CD3FC1          CALL DSPMSG           ; display result
              ;
C13D 18C1            JR   START
              ;
              ; display message subroutine
              ;
C13F 7E       DSPMSG:LD   A,(HL)
C140 A7              AND  A
C141 C8              RET  Z
C142 CD5702          CALL 0257H            ; display a character
```

```
C145 23               INC   HL
C146 18F7             JR    DSPMSG
                ;
                ; message area
                ;
C148 0D0A0A20 PRMPT1:DEFB 0DH,0AH,0AH,'      x%    ? ',0
C14C 20202020
C150 78252020
C154 203F2000
C158 20202020 PRMPT2:DEFB '      y%    ? ',0
C15C 20792520
C160 20203F20
C164 00
C165 7825202B ANSWER:DEFB 'x% + y% --> ',0
C169 20792520
C16D 2D2D3E20
C171 00
                ;
C172                  END


*GC100


      x%    ? 20
      y%    ? 30
x% + y% -->  50

      x%    ? 500
      y%    ? -600
x% + y% --> -100

      x%    ? 10
      y%    ? 0.5
x% + y% -->  10

      x%    ? 300
      y%    ?
x% + y% -->  300

      x%    ?

*
```

２８ＦＤＨ　整数の乗算

アドレス　２８ＦＤＨ
機　　能　整数の乗算
レジスタ　ＡＦＢＣＤＥＨＬ

解　　説　レジスタＤＥに格納された有符号１６ビットの整数を被乗数とし，レジスタＨＬに格納された有符号１６ビットの整数を乗数として，両者の間で乗算を行い，演算結果をレジスタＨＬおよびフローティング・アキュムレータ（Ｆ０Ａ８Ｈ～Ｆ０Ａ９Ｈ番地）に格納してもどります．

演算結果が整数型として扱い得る範囲を越えた場合には，単精度の値としてフローティング・アキュムレータに返されますが，その場合にはレジスタＨＬに与えられたパラメータは意味を持ちません．

サンプル

```
                 ;
                 ; --- multiplication for integer ---
                 ;
                         ORG   0C100H
                 ;
C100 2148C1      START:  LD    HL,PRMPT1
C103 CD3FC1              CALL  DSPMSG
C106 CD7E1B              CALL  1B7EH             ; screen editor
C109 DA665C              JP    C,5C66H           ; system monitor if stop key
C10C 23                  INC   HL
C10D CDBE2B              CALL  2BBEH             ; convert into binary code
C110 CD7F27              CALL  277FH             ; convert into integer
C113 ED5BA8F0            LD    DE,(0F0A8H)       ; store 1st argument into reg-de
C117 D5                  PUSH  DE
                 ;
C118 2158C1              LD    HL,PRMPT2
C11B CD3FC1              CALL  DSPMSG
C11E CD7E1B              CALL  1B7EH             ; screen editor
C121 23                  INC   HL
C122 CDBE2B              CALL  2BBEH             ; convert into binary code
C125 CD7F27              CALL  277FH             ; convert into integer
C128 2AA8F0              LD    HL,(0F0A8H)       ; set 2nd argument into reg-hl
                 ;
C12B D1                  POP   DE                ; set 1st argument into reg-de
                 ;
C12C CDFD28              CALL  28FDH             ;*multiplication for integer
                 ;
C12F CD222D              CALL  2D22H             ; convert into character code
C132 E5                  PUSH  HL
C133 2165C1              LD    HL,ANSWER
C136 CD3FC1              CALL  DSPMSG
C139 E1                  POP   HL
C13A CD3FC1              CALL  DSPMSG            ; display result
                 ;
C13D 18C1                JR    START
                 ;
                 ; display message subroutine
                 ;
C13F 7E          DSPMSG:LD     A,(HL)
C140 A7                  AND   A
C141 C8                  RET   Z
C142 CD5702              CALL  0257H             ; display a character
```

```
C145 23                 INC   HL
C146 18F7               JR    DSPMSG
               ;
               ; message area
               ;
C148 0D0A0A20 PRMPT1:DEFB 0DH,0AH,0AH,'      x%    ? ',0
C14C 20202020
C150 78252020
C154 203F2000
C158 20202020 PRMPT2:DEFB '      y%    ? ',0
C15C 20792520
C160 20203F20
C164 00
C165 7825202A ANSWER:DEFB 'x% * y% --> ',0
C169 20792520
C16D 2D2D3E20
C171 00
               ;
C172                    END


*GC100


      x%    ? 1000
      y%    ? 123
x% * y% -->  123000

      x%    ? 123.456
      y%    ? 789
x% * y% -->  97047

      x%    ? -500
      y%    ? 50
x% * y% --> -25000

      x%    ? 1000
      y%    ? 60000
Overflow
Ok
```

２９５０Ｈ　整数の除算１

アドレス	２９５０Ｈ
機　能	整数の除算１
レジスタ	ＡＦＢＣＤＥＨＬ

解　説　レジスタＤＥに格納された有符号１６ビットの整数を被除数とし，レジスタＨＬに格納された有符号１６ビットの整数を除数として，両者の間で除算を行い，演算結果をレジスタＨＬおよびフローティング・アキュムレータ（Ｆ０Ａ８Ｈ～Ｆ０Ａ９Ｈ番地）に格納してもどります．

ただし，レジスタＨＬに与える除数が，００００Ｈ（ゼロ）であった場合にのみ，“Division by zero”エラーが発生しインタプリタの制御下に入ります．

サンプル

```
                ;
                ; --- division for integer ---
                ;
                        ORG   0C100H
                ;
C100 2148C1     START:  LD    HL,PRMPT1
C103 CD3FC1             CALL  DSPMSG
C106 CD7E1B             CALL  1B7EH          ; screen editor
C109 DA665C             JP    C,5C66H        ; system monitor if stop key
C10C 23                 INC   HL
C10D CDBE2B             CALL  2BBEH          ; convert into binary code
C110 CD7F27             CALL  277FH          ; convert into integer
C113 ED5BA8F0           LD    DE,(0F0A8H)    ; store 1st argument into reg-de
C117 D5                 PUSH  DE
                ;
C118 2158C1             LD    HL,PRMPT2
C11B CD3FC1             CALL  DSPMSG
C11E CD7E1B             CALL  1B7EH          ; screen editor
C121 23                 INC   HL
C122 CDBE2B             CALL  2BBEH          ; convert into binary code
C125 CD7F27             CALL  277FH          ; convert into integer
C128 2AA8F0             LD    HL,(0F0A8H)    ; set 2nd argument into reg-hl
                ;
C12B D1                 POP   DE             ; set 1st argument into reg-de
                ;
C12C CD5029             CALL  2950H          ;*division for integer
                ;
C12F CD222D             CALL  2D22H          ; convert into character code
C132 E5                 PUSH  HL
C133 2165C1             LD    HL,ANSWER
C136 CD3FC1             CALL  DSPMSG
C139 E1                 POP   HL
C13A CD3FC1             CALL  DSPMSG         ; display result
                ;
C13D 18C1               JR    START
                ;
                ; display message subroutine
                ;
C13F 7E         DSPMSG:LD     A,(HL)
C140 A7                 AND   A
C141 C8                 RET   Z
C142 CD5702             CALL  0257H          ; display a character
C145 23                 INC   HL
C146 18F7               JR    DSPMSG
```

```
                ;
                ; message area
                ;
C148 0D0A0A20 PRMPT1:DEFB 0DH,0AH,0AH,'      x%   ? ',0
C14C 20202020
C150 78252020
C154 203F2000
C158 20202020 PRMPT2:DEFB '      y%   ? ',0
C15C 20792520
C160 20203F20
C164 00
C165 7825202F ANSWER:DEFB 'x% / y% --> ',0
C169 20792520
C16D 2D2D3E20
C171 00
                ;
C172                   END


*GC100


     x%   ? 20
     y%   ? 6
x% / y% -->  3

     x%   ? 23.456
     y%   ? 10
x% / y% -->  2

     x%   ? -500
     y%   ? 20
x% / y% --> -25

     x%   ? 100
     y%   ? 1
x% / y% -->  100

     x%   ? 100
     y%   ? 0.5
Division by zero
Ok
```

299DH　整数の符号反転1

アドレス	299DH
機　　能	整数の符号反転1
レジスタ	AFBCDEHL

解　　説　フローティング・アキュムレータに格納された有符号16ビットの整数の符号を反転（NEGATION）し，演算結果をレジスタHLおよび整数型のフローティング・アキュムレータに格納してもどります.

ただし，フローティング・アキュムレータに格納されていたデータが8000H（－32768）であった場合には，符号反転の結果（32768）が整数型として扱うことのできる範囲を超えてしまうため，演算結果は単精度型の実数としてフローティング・アキュムレータだけに格納されます.

具体的には，整数型フローティング・アキュムレータ（F0A8H～F0A9H番地）のデータをレジスタHLに移動後29A0H番地にエントリします.

29A0H　整数の符号反転2

アドレス	29A0H
機　　能	整数の符号反転2
レジスタ	AFBCDEHL

解　　説　レジスタHLに格納された有符号16ビットの整数の符号を反転（NEGATION）し，演算結果をレジスタHLおよび整数型のフローティング・アキュムレータに格納してもどります.

ただし，レジスタHLに格納されていたデータが8000H（－32768）であった場合には，符号反転の結果（32768）が整数型として扱うことのできる範囲を超えてしまうため，演算結果は単精度型の実数としてフローティング・アキュムレータだけに格納されます.

サンプル

```
                ;
                ; --- negation for integer ---
                ;
                        ORG   0C100H
                ;
C100 212FC1     START: LD     HL,PROMPT
C103 CD26C1            CALL  DSPMSG
C106 CD7E1B            CALL  1B7EH           ; screen editor
C109 DA665C            JP    C,5C66H         ; jump to system monitor if stop key
C10C 23                INC   HL
C10D CDBE2B            CALL  2BBEH           ; convert into binary code
C110 CD7F27            CALL  277FH           ; convert into integer
                ;
C113 CD9D29            CALL  299DH           ;*negation for integer in facc
                ;
C116 CD222D            CALL  2D22H           ; convert into character code
C119 E5                PUSH  HL
C11A 213BC1            LD    HL,ANSWER
C11D CD26C1            CALL  DSPMSG
C120 E1                POP   HL
C121 CD26C1            CALL  DSPMSG          ; display result
                ;
C124 18DA              JR    START
                ;
                ; display message subroutine
                ;
C126 7E         DSPMSG:LD    A,(HL)
C127 A7                AND   A
C128 C8                RET   Z
C129 CD5702            CALL  0257H           ; display a character
C12C 23                INC   HL
C12D 18F7              JR    DSPMSG
                ;
                ; message area
                ;
C12F 0D0A0A20   PROMPT:DEFB 0DH,0AH,0AH,' x%   ? ',0
C133 78252020
C137 203F2000
C13B 2D782520   ANSWER:DEFB '-x% --> ',0
C13F 2D2D3E20
C143 00
                ;
C144                   END


*GC100


 x%   ? &h8000
-x% -->  32768

 x%   ? &h7fff
-x% --> -32767

 x%   ? 1234
-x% --> -1234

 x%   ? -22222
-x% -->  22222

 x%   ? 0.005
-x% -->  0

 x%   ? 40000
Overflow
Ok
```

29B2H　整数の剰余演算

アドレス　29B2H

機　能　整数の剰余演算

レジスタ　AFBCDEHL

解　説　レジスタDEに格納された有符号16ビットの整数を被除数とし，レジスタHLに格納された有符号16ビットの整数を除数として，両者の間で除算を行い，剰余をレジスタDEおよびフローティング・アキュムレータ（F0A8H～F0A9H番地）に格納してもどります．

ただし，レジスタHLに与える除数が，0000H（ゼロ）であった場合にのみ“Division by zero”エラーが発生し，インタプリタの制御下に入ります．

サンプル

```
                ;
                ; --- calculate remainder ---
                ;
                        ORG  0C100H
                ;
C100 2148C1     START:  LD   HL,PRMPT1
C103 CD3FC1             CALL DSPMSG
C106 CD7E1B             CALL 1B7EH           ; screen editor
C109 DA665C             JP   C,5C66H         ; jump to system monitor if stop key
C10C 23                 INC  HL
C10D CDBE2B             CALL 2BBEH           ; convert into binary code
C110 CD7F27             CALL 277FH           ; convert into integer
C113 ED5BA8F0           LD   DE,(0F0A8H)     ; store 1st argument into reg-de
C117 D5                 PUSH DE
                ;
C118 215AC1             LD   HL,PRMPT2
C11B CD3FC1             CALL DSPMSG
C11E CD7E1B             CALL 1B7EH           ; screen editor
C121 23                 INC  HL
C122 CDBE2B             CALL 2BBEH           ; convert into binary code
C125 CD7F27             CALL 277FH           ; convert into integer
C128 2AA8F0             LD   HL,(0F0A8H)     ; set 2nd argument into reg-hl
                ;
C12B D1                 POP  DE              ; set 1st argument into reg-de
                ;
C12C CDB229             CALL 29B2H           ;*calculate remainder
                ;
C12F CD222D             CALL 2D22H           ; convert into character code
C132 E5                 PUSH HL
C133 2169C1             LD   HL,ANSWER
C136 CD3FC1             CALL DSPMSG
C139 E1                 POP  HL
C13A CD3FC1             CALL DSPMSG          ; display result
                ;
C13D 18C1               JR   START
                ;
                ; display message subroutine
                ;
                ;    destroys : register a,f,h,l
                ;
C13F 7E         DSPMSG:LD    A,(HL)
C140 A7                 AND  A
C141 C8                 RET  Z
```

```
C142 CD5702          CALL 0257H            ; display a character
C145 23              INC  HL
C146 18F7            JR   DSPMSG
               ;
               ; message area
               ;
C148 0D0A0A20 PRMPT1:DEFB 0DH,0AH,0AH,'       x%    ? ',0
C14C 20202020
C150 20207825
C154 2020203F
C158 2000
C15A 20202020 PRMPT2:DEFB '       y%    ? ',0
C15E 20202079
C162 25202020
C166 3F2000
C169 72656D61 ANSWER:DEFB 'remainder --> ',0
C16D 696E6465
C171 72202D2D
C175 3E2000
               ;
C178                 END


*GC100


       x%    ? 10
       y%    ? 3
remainder -->  1

       x%    ? 1000
       y%    ? 501
remainder -->  499

       x%    ? 10
       y%    ? -3
remainder -->  1

       x%    ? 1000
       y%    ? -501
remainder -->  499

       x%    ? -10000
       y%    ? -3
remainder --> -1

       x%    ?
*
```

29C3H　倍精度型実数の減算

アドレス	29C3H
機　能	倍精度型実数の減算
レジスタ	AFBCDEHL

解　説　フローティング・アキュムレータ（F0A4H～F0ABH番地）に格納された倍精度型の実数を被減数とし，サブ・フローティング・アキュムレータ（F0AEH～F0B5H番地）に格納された倍精度型の実数を減数として，両者の間で減算を行い，演算結果をフローティング・アキュムレータに格納してもどります．

サンプル

```
                ;
                ; --- subtraction for double precision ---
                ;
                        ORG   0C100H
                ;
C100 2163C1     START:  LD    HL,PRMPT1
C103 CD5AC1             CALL  DSPMSG
C106 CD7E1B             CALL  1B7EH           ; screen editor
C109 DA665C             JP    C,5C66H         ; system monitor if stop key
C10C 23                 INC   HL
C10D CDBE2B             CALL  2BBEH           ; convert into binary code
C110 CDDF27             CALL  27DFH           ; convert into double precision
                ;
C113 21A4F0             LD    HL,0F0A4H       ;
C116 118DC1             LD    DE,BUFFER       ; store 1st argument into buffer
C119 010800             LD    BC,8            ;
C11C EDB0               LDIR
                ;
C11E 2173C1             LD    HL,PRMPT2
C121 CD5AC1             CALL  DSPMSG
C124 CD7E1B             CALL  1B7EH           ; screen editor
C127 DA665C             JP    C,5C66H         ; system monitor if stop key
C12A 23                 INC   HL
C12B CDBE2B             CALL  2BBEH           ; convert into binary code
C12E CDDF27             CALL  27DFH           ; convert into double precision
                ;
C131 21A4F0             LD    HL,0F0A4H       ;
C134 11AEF0             LD    DE,0F0AEH       ; set 2nd argument into sub-facc
C137 010800             LD    BC,8            ;
C13A EDB0               LDIR
                ;
C13C 218DC1             LD    HL,BUFFER       ;
C13F 11A4F0             LD    DE,0F0A4H       ; set 1st argument into facc
C142 010800             LD    BC,8            ;
C145 EDB0               LDIR
                ;
C147 CDC329             CALL  29C3H           ;*subtraction for double precision
                ;
C14A CD222D             CALL  2D22H           ; convert into character code
C14D E5                 PUSH  HL
C14E 2180C1             LD    HL,ANSWER
C151 CD5AC1             CALL  DSPMSG
C154 E1                 POP   HL
C155 CD5AC1             CALL  DSPMSG          ; display result
                ;
C158 18A6               JR    START
```

```
                ;
                ; display message subroutine
                ;
C15A 7E         DSPMSG:LD   A,(HL)
C15B A7                AND  A
C15C C8                RET  Z
C15D CD5702            CALL 0257H             ; display a character
C160 23                INC  HL
C161 18F7              JR   DSPMSG
                ;
                ; message area
                ;
C163 0D0A0A20 PRMPT1:DEFB 0DH,0AH,0AH,'      x#    ? ',0
C167 20202020
C16B 78232020
C16F 203F2000
C173 20202020 PRMPT2:DEFB '      y#    ? ',0
C177 20792320
C17B 20203F20
C17F 00
C180 7823202D ANSWER:DEFB 'x# - y# --> ',0
C184 20792320
C188 2D2D3E20
C18C 00
                ;
                ; buffer for the first argument
                ;
C18D            BUFFER:DEFS 8
                ;
C195                   END


*GC100


      x#    ? 1
      y#    ? 100000000000000
x# - y# --> -99999999999999

      x#    ? 1
      y#    ? 1000000000000000000000000
x# - y# --> -1D+24

      x#    ? 300
      y#    ? 1.5
x# - y# -->  298.5

      x#    ? 10000000000000000000000000000000000
      y#    ? 30000000000000000000000000000000000
x# - y# --> -2D+34

      x#    ? 1
      y#    ? 300000000000000000000000000000000000000000000000000000000000000
Overflow
Ok
```

2 9 C A H　倍精度型実数の加算

アドレス	2 9 C A H
機　　能	倍精度型実数の加算
レジスタ	A F B C D E H L

解　　説　フローティング・アキュムレータ（F 0 A 4 H～F 0 A B H番地）に格納された倍精度型の実数を被加数とし，サブ・フローティング・アキュムレータ（F 0 A E H～F 0 B 5 H番地）に格納された倍精度型の実数を加数として，両者の間で加算を行い，演算結果をフローティング・アキュムレータに格納してもどります．

サンプル

```
                  ;
                  ; --- addition for double precision ---
                  ;
                          ORG   0C100H
                  ;
C100 2163C1       START:  LD    HL,PRMPT1
C103 CD5AC1               CALL  DSPMSG
C106 CD7E1B               CALL  1B7EH               ; screen editor
C109 DA665C               JP    C,5C66H             ; system monitor if stop key
C10C 23                   INC   HL
C10D CDBE2B               CALL  2BBEH               ; convert into binary code
C110 CDDF27               CALL  27DFH               ; convert into double precision
                  ;
C113 21A4F0               LD    HL,0F0A4H           ;
C116 118DC1               LD    DE,BUFFER           ; store 1st argument into buffer
C119 010800               LD    BC,8                ;
C11C EDB0                 LDIR
                  ;
C11E 2173C1               LD    HL,PRMPT2
C121 CD5AC1               CALL  DSPMSG
C124 CD7E1B               CALL  1B7EH               ; screen editor
C127 DA665C               JP    C,5C66H             ; system monitor if stop key
C12A 23                   INC   HL
C12B CDBE2B               CALL  2BBEH               ; convert into binary code
C12E CDDF27               CALL  27DFH               ; convert into double precision
                  ;
C131 21A4F0               LD    HL,0F0A4H
C134 11AEF0               LD    DE,0F0AEH
C137 010800               LD    BC,8
C13A EDB0                 LDIR
                  ;
C13C 218DC1               LD    HL,BUFFER
C13F 11A4F0               LD    DE,0F0A4H
C142 010800               LD    BC,8
C145 EDB0                 LDIR
                  ;
C147 CDCA29               CALL  29CAH               ;*addition for double precision
                  ;
C14A CD222D               CALL  2D22H               ; convert into character code
C14D E5                   PUSH  HL
C14E 2180C1               LD    HL,ANSWER
C151 CD5AC1               CALL  DSPMSG
C154 E1                   POP   HL
C155 CD5AC1               CALL  DSPMSG              ; display result
                  ;
C158 18A6                 JR    START
```

```
                ;
                ; display message subroutine
                ;
C15A 7E         DSPMSG:LD   A,(HL)
C15B A7                AND  A
C15C C8                RET  Z
C15D CD5702            CALL 0257H           ; display a character
C160 23                INC  HL
C161 18F7              JR   DSPMSG
                ;
                ; message area
                ;
C163 0D0A0A20 PRMPT1:DEFB 0DH,0AH,0AH,'      x#    ? ',0
C167 20202020
C16B 78232020
C16F 203F2000
C173 20202020 PRMPT2:DEFB '      y#    ? ',0
C177 20792320
C17B 20203F20
C17F 00
C180 7823202B ANSWER:DEFB 'x# + y# --> ',0
C184 20792320
C188 2D2D3E20
C18C 00
                ;
                ; buffer for the first argument
                ;
C18D            BUFFER:DEFS 8
                ;
C195                   END


*GC100


      x#    ? 12345678
      y#    ? 1234567800000000
x# + y# -->  1234567812345678

      x#    ? 1
      y#    ? 2
x# + y# -->  3

      x#    ? 0.00000000000000001
      y#    ? 0.00000000000000001
x# + y# -->  1.999999802114196D-17

      x#    ? 0.000000000000000000000000000000000000001
      y#    ? 0
x# + y# -->  1.000000005110564D-38

      x#    ? 1.5
      y#    ? 1.5
x# + y# -->  3

*
```

2AF4H　倍精度型実数の乗算

アドレス	2AF4H
機　　能	倍精度型実数の乗算
レジスタ	AFBCDEHL

解　　説　フローティング・アキュムレータ（F0A4H～F0ABH番地）に格納された倍精度型の実数を被乗数とし，サブ・フローティング・アキュムレータ（F0AEH～F0B5H番地）に格納された倍精度型の実数を乗数として，両者の間で乗算を行い，演算結果をフローティング・アキュムレータに格納してもどります．

サンプル

```
               ;
               ; --- multiplication for double precision ---
               ;
                      ORG   0C100H
               ;
C100 2163C1    START: LD    HL,PRMPT1
C103 CD5AC1           CALL  DSPMSG
C106 CD7E1B           CALL  1B7EH          ; screen editor
C109 DA665C           JP    C,5C66H        ; system monitor if stop key
C10C 23               INC   HL
C10D CDBE2B           CALL  2BBEH          ; convert into binary code
C110 CDDF27           CALL  27DFH          ; convert into double precision
               ;
C113 21A4F0           LD    HL,0F0A4H      ;
C116 118DC1           LD    DE,BUFFER      ; store 1st argument into buffer
C119 010800           LD    BC,8           ;
C11C EDB0             LDIR
               ;
C11E 2173C1           LD    HL,PRMPT2
C121 CD5AC1           CALL  DSPMSG
C124 CD7E1B           CALL  1B7EH          ; screen editor
C127 DA665C           JP    C,5C66H        ; system monitor if stop key
C12A 23               INC   HL
C12B CDBE2B           CALL  2BBEH          ; convert into binary code
C12E CDDF27           CALL  27DFH          ; convert into double precision
               ;
C131 21A4F0           LD    HL,0F0A4H      ;
C134 11AEF0           LD    DE,0F0AEH      ; set 2nd argument into sub-facc
C137 010800           LD    BC,8           ;
C13A EDB0             LDIR
               ;
C13C 218DC1           LD    HL,BUFFER      ;
C13F 11A4F0           LD    DE,0F0A4H      ; set 1st argument into facc
C142 010800           LD    BC,8           ;
C145 EDB0             LDIR
               ;
C147 CDF42A           CALL  2AF4H          ;*multiplication for double
               ;
C14A CD222D           CALL  2D22H          ; convert into character code
C14D E5               PUSH  HL
C14E 2180C1           LD    HL,ANSWER
C151 CD5AC1           CALL  DSPMSG
C154 E1               POP   HL
C155 CD5AC1           CALL  DSPMSG         ; display result
               ;
C158 18A6             JR    START
```

```
                ;
                ; display message subroutine
                ;
C15A 7E         DSPMSG:LD   A,(HL)
C15B A7                AND  A
C15C C8                RET  Z
C15D CD5702            CALL 0257H              ; display a character
C160 23                INC  HL
C161 18F7              JR   DSPMSG
                ;
                ; message area
                ;
C163 0D0A0A20 PRMPT1:DEFB 0DH,0AH,0AH,'      x#   ? ',0
C167 20202020
C16B 78232020
C16F 203F2000
C173 20202020 PRMPT2:DEFB '      y#   ? ',0
C177 20792320
C17B 20203F20
C17F 00
C180 7823202A ANSWER:DEFB 'x# * y# --> ',0
C184 20792320
C188 2D2D3E20
C18C 00
                ;
                ; buffer for the first argument
                ;
C18D            BUFFER:DEFS 8
                ;
C195                   END


*GC100


     x#   ? 100000
     y#   ? 100000
x# * y# -->  10000000000

     x#   ? 1.000123456789
     y#   ? 100000000000
x# * y# -->  100012345678.9

     x#   ? 3
     y#   ? -0.5
x# * y# --> -1.5

     x#   ? 0
     y#   ? 0.0
x# * y# -->  0

     x#   ?
     y#   ?

*
```

2B37H　倍精度型実数の除算

アドレス	2B37H
機　　能	倍精度型実数の除算
レジスタ	AFBCDEHL

解　　説　フローティング・アキュムレータ（F0A4H～F0ABH番地）に格納された倍精度型の実数を被除数とし，サブ・フローティング・アキュムレータ（F0AEH～F0B5H番地）に格納された倍精度型の実数を除数として，両者の間で除算を行い，演算結果をフローティング・アキュムレータに格納してもどります．

もし，サブ・フローティング・アキュムレータに与える除数が0（ゼロ）であった場合には，“Division by zero”エラーが発生しインタプリタの制御下に入ります．

サンプル

```
                 ;
                 ; --- division for double precision ---
                 ;
                         ORG   0C100H
                 ;
C100 2163C1      START:  LD    HL,PRMPT1
C103 CD5AC1              CALL  DSPMSG
C106 CD7E1B              CALL  1B7EH            ; screen editor
C109 DA665C              JP    C,5C66H          ; system monitor if sotp key
C10C 23                  INC   HL
C10D CDBE2B              CALL  2BBEH            ; convert into binary code
C110 CDDF27              CALL  27DFH            ; convert into double precision
                 ;
C113 21A4F0              LD    HL,0F0A4H        ;
C116 118DC1              LD    DE,BUFFER        ; store 1st argument into buffer
C119 010800              LD    BC,8             ;
C11C EDB0                LDIR
                 ;
C11E 2173C1              LD    HL,PRMPT2
C121 CD5AC1              CALL  DSPMSG
C124 CD7E1B              CALL  1B7EH            ; screen editor
C127 DA665C              JP    C,5C66H          ; system monitor if stop key
C12A 23                  INC   HL
C12B CDBE2B              CALL  2BBEH            ; convert into binary code
C12E CDDF27              CALL  27DFH            ; convert into double precision
                 ;
C131 21A4F0              LD    HL,0F0A4H        ;
C134 11AEF0              LD    DE,0F0AEH        ; set 2nd argument into sub-facc
C137 010800              LD    BC,8             ;
C13A EDB0                LDIR
                 ;
C13C 218DC1              LD    HL,BUFFER        ;
C13F 11A4F0              LD    DE,0F0A4H        ; set 1st argument into facc
C142 010800              LD    BC,8             ;
C145 EDB0                LDIR
                 ;
C147 CD372B              CALL  2B37H            ;*division for double precision
                 ;
C14A CD222D              CALL  2D22H            ; convert into character code
C14D E5                  PUSH  HL
```

```
C14E 2180C1              LD   HL,ANSWER
C151 CD5AC1              CALL DSPMSG
C154 E1                  POP  HL
C155 CD5AC1              CALL DSPMSG             ; display result
                ;
C158 18A6                JR   START
                ;
                ; display message subroutine
                ;
C15A 7E         DSPMSG:LD   A,(HL)
C15B A7                  AND  A
C15C C8                  RET  Z
C15D CD5702              CALL 0257H              ; display a character
C160 23                  INC  HL
C161 18F7                JR   DSPMSG
                ;
                ; message area
                ;
C163 0D0A0A20 PRMPT1:DEFB 0DH,0AH,0AH,'      x#   ? ',0
C167 20202020
C16B 78232020
C16F 203F2000
C173 20202020 PRMPT2:DEFB '      y#   ? ',0
C177 20792320
C17B 20203F20
C17F 00
C180 7823202F ANSWER:DEFB 'x# / y# --> ',0
C184 20792320
C188 2D2D3E20
C18C 00
                ;
                ; buffer for the first argument
                ;
C18D            BUFFER:DEFS 8
                ;
C195                    END


*GC100


     x#   ? 20
     y#   ? 6
x# / y# -->  3.333333333333333

     x#   ? 20
     y#   ? 3
x# / y# -->  6.666666666666667

     x#   ? 1.23456789012345
     y#   ? 1000
x# / y# -->  1.23456789012345D-03

     x#   ? 300
     y#   ? 300.1
x# / y# -->  .999666757409258

     x#   ? 5
     y#   ? 0
Division by zero
Ok
```

2BB7H　文字列から倍精度実数型データへの変換

アドレス	2BB7H
機　　能	文字列から倍精度実数型データへの変換
レジスタ	AFBCDEHL

解　　説　レジスタHLによって指定するアドレスから格納されている，キャラクタ・コードによる数値データを，倍精度実数型のバイナリ・コードに変換しフローティング・アキュムレータ（FACC）に格納してもどります．

通常のデータであれば，すべて区別なく倍精度実数型として格納され，EF45H番地にも倍精度実数型を示す08Hが入りますが，次に掲げる諸条件のいずれかを満たすデータのみは，整数型または単精度実数型の数値として変換されます．

1）先頭に“&O”または“&”を伴った8進形式のデータ
2）先頭に“&H”を伴った16進形式のデータ
3）末尾に“%”を伴った整数型のデータ
4）末尾に“！”を伴った単精度実数型のデータ
5）“E”を伴った指数形式のデータ

上記の条件がいずれも満たされていなければ，たとえば“10”であっても整数型とは見なされません．“10%”であれば整数型となります．

また，データの中に数値に関係のないキャラクタ・コード（00Hを含む）が存在する場合には，直前のキャラクタまでが有効となります．

サンプル

```
                ;
                ; --- input and display number ---
                ;
                        ORG   0C100H
                ;
C100 2140C1     START:  LD    HL,PROMPT
C103 CD37C1             CALL  DSPMSG
C106 CD7E1B             CALL  1B7EH             ; screen editor
C109 DA665C             JP    C,5C66H           ; jump to system monitor if stop key
C10C 23                 INC   HL
C10D CDB72B             CALL  2BB7H             ;*into double precision binary code
                ;
C110 3A45EF             LD    A,(0EF45H)        ; get type of floating accumulator
                ;
C113 FE02               CP    02H
C115 2005               JR    NZ,NOT2
C117 214FC1             LD    HL,TYPE2          ; integer
C11A 1810               JR    DSPLAY
                ;
C11C FE04       NOT2:   CP    04H
C11E 2005               JR    NZ,NOT4
C120 215BC1             LD    HL,TYPE4          ; single precision number
```

```
C123 1807              JR   DSPLAY
               ;
C125 FE08      NOT4:   CP   08H
C127 20D7              JR   NZ,START            ; input a number again if error
C129 2177C1            LD   HL,TYPE8            ; double precision number
               ;
C12C CD37C1    DSPLAY:CALL DSPMSG
               ;
C12F CD222D            CALL 2D22H               ; convert into character code
C132 CD37C1            CALL DSPMSG
               ;
C135 18C9              JR   START
               ;
               ; display message subroutine
               ;
               ;    destroys : register a,f,h,l
               ;
C137 7E        DSPMSG:LD    A,(HL)
C138 A7                AND  A
C139 C8                RET  Z
C13A CD5702            CALL 0257H               ; display a character
C13D 23                INC  HL
C13E 18F7              JR   DSPMSG
               ;
               ; message area
               ;
C140 0D0A0A61 PROMPT:DEFB 0DH,0AH,0AH,'a number : ',0
C144 206E756D
C148 62657220
C14C 3A2000
C14F 696E7465 TYPE2: DEFB 'integer of ',0
C153 67657220
C157 6F662000
C15B 73696E67 TYPE4: DEFB 'single precision number of ',0
C15F 6C652070
C163 72656369
C167 73696F6E
C16B 206E756D
C16F 62657220
C173 6F662000
C177 646F7562 TYPE8: DEFB 'double precision number of ',0
C17B 6C652070
C17F 72656369
C183 73696F6E
C187 206E756D
C18B 62657220
C18F 6F662000
               ;
C193                   END


*GC100


a number : 333
double precision number of  333

a number : 3.3
double precision number of  3.3

a number : 3.3!
single precision number of  3.3

a number : &h55
integer of  85

a number : &O55
integer of  45

a number : 3E+5
single precision number of  300000

a number :

*
```

2BBEH　文字列から数値データへの変換

アドレス	2BBEH
機　能	文字列から数値データへの変換
レジスタ	AFBCDE－－

解　説　レジスタHLによって指定するアドレスから格納されている，キャラクタ・コードによる数値データを，倍精度実数型，単精度実数型，または整数型のバイナリ・コードに変換してフローティング・アキュムレータ（FACC）に格納し，EF45H番地には型を示す数値（08H，04H，または02H）を格納します．

データの中に数値に関係のないキャラクタ・コード（00Hを含む）が存在する場合には，直前のキャラクタまでが有効となります．

サンプル

```
                ;
                ; --- input and display number ---
                ;
                        ORG   0C100H
                ;
C100 2140C1     START:  LD    HL,PROMPT
C103 CD37C1             CALL  DSPMSG
C106 CD7E1B             CALL  1B7EH              ; screen editor
C109 DA665C             JP    C,5C66H            ; jump to system monitor if stop key
C10C 23                 INC   HL
C10D CDBE2B             CALL  2BBEH              ;*convert into binary code
                ;
C110 3A45EF             LD    A,(0EF45H)         ; get type of floating accumulator
                ;
C113 FE02               CP    02H
C115 2005               JR    NZ,NOT2
C117 214FC1             LD    HL,TYPE2           ; integer
C11A 1810               JR    DSPLAY
                ;
C11C FE04       NOT2:   CP    04H
C11E 2005               JR    NZ,NOT4
C120 215BC1             LD    HL,TYPE4           ; single precision number
C123 1807               JR    DSPLAY
                ;
C125 FE08       NOT4:   CP    08H
C127 20D7               JR    NZ,START           ; input a number again if error
C129 2177C1             LD    HL,TYPE8           ; double precision number
                ;
C12C CD37C1     DSPLAY:CALL DSPMSG
                ;
C12F CD222D             CALL  2D22H              ; convert into character code
C132 CD37C1             CALL  DSPMSG
                ;
C135 18C9               JR    START
                ;
                ; display message subroutine
                ;
C137 7E         DSPMSG:LD    A,(HL)
C138 A7                 AND   A
C139 C8                 RET   Z
C13A CD5702             CALL  0257H              ; display a character
C13D 23                 INC   HL
```

```
C13E 18F7            JR    DSPMSG
                ;
                ; message area
                ;
C140 0D0A0A61 PROMPT:DEFB 0DH,0AH,0AH,'a number : ',0
C144 206E756D
C148 62657220
C14C 3A2000
C14F 696E7465 TYPE2: DEFB 'integer of ',0
C153 67657220
C157 6F662000
C15B 73696E67 TYPE4: DEFB 'single precision number of ',0
C15F 6C652070
C163 72656369
C167 73696F6E
C16B 206E756D
C16F 62657220
C173 6F662000
C177 646F7562 TYPE8: DEFB 'double precision number of ',0
C17B 6C652070
C17F 72656369
C183 73696F6E
C187 206E756D
C18B 62657220
C18F 6F662000
                ;
C193                 END


*GC100


a number : 333
integer of  333

a number : 333.0
single precision number of  333

a number : 33333333333333333333333333333333
double precision number of  3.333333333333333D+31

a number : 3#
double precision number of  3

a number : z80
integer of  0

a number : 8086!
double precision number of  8086

a number : 333333333333333333333333333333333333333333333333
Overflow
Ok
```

2D0BH “ in ”と行番号の出力

アドレス　2D0BH

機　能　“ in ”と行番号の出力

レジスタ　AFBCDEHL

解　説　CRT画面上の現カーソル位置から，メッセージ“ in ”を表示後，さらにレジスタHLに格納された16ビットのデータを無符号の10進整数として表示します．

インタプリタでは，プログラム中断（ブレーク）時，エラー発生時等の，発生行番号を表示するために，このシステム・サブルーチンを使用しています．

また，EB49H番地の出力フラグが01H～7FHの場合には，プリンタへの出力を行います．

サンプル

```
               ;
               ; --- increment line number ---
               ;
                       ORG  0C100H
               ;
C100 210000            LD   HL,0             ; initialize line number
               ;
C103 22A8F0    LOOP:   LD   (0F0A8H),HL      ;
C106 3E02              LD   A,2              ; store line number into facc
C108 3245EF            LD   (0EF45H),A       ;
               ;
C10B E5                PUSH HL               ;
C10C CD132D            CALL 2D13H            ; display line number
C10F E1                POP  HL               ;
               ;
C110 114AC1            LD   DE,MESAGE
C113 CD41C1            CALL DSPMSG
C116 CD37C1            CALL WAIT
               ;
C119 DB09              IN   A,(09H)          ;
C11B CB47              BIT  0,A              ; exit if stop key
C11D 2806              JR   Z,EXIT           ;
               ;
C11F 110A00            LD   DE,10
C122 19                ADD  HL,DE
               ;
C123 18DE              JR   LOOP
               ;
C125 1163C1    EXIT:   LD   DE,BREAK1
C128 CD41C1            CALL DSPMSG
               ;
C12B CD0B2D            CALL 2D0BH            ; display line number
               ;
C12E 1168C1            LD   DE,BREAK2
C131 CD41C1            CALL DSPMSG
               ;
C134 C3665C            JP   5C66H            ; jump to system monitor
               ;
               ; wait subroutine
               ;
               ;    destroys : register f,b,c
               ;
```

```
C137 0EFF      WAIT:  LD   C,0FFH
C139 06FF      WAIT1: LD   B,0FFH
C13B 10FE      WAIT2: DJNZ WAIT2
C13D 0D               DEC  C
C13E 20F9             JR   NZ,WAIT1
C140 C9               RET
               ;
               ; display message subroutine
               ;
               ;    destroys : register a,f,d,e
               ;
C141 1A        DSPMSG:LD   A,(DE)
C142 A7               AND  A
C143 C8               RET  Z
C144 CD5702           CALL 0257H            ; display a character
C147 13               INC  DE
C148 18F7             JR   DSPMSG
               ;
               ; message area
               ;
C14A 20706C65 MESAGE:DEFB ' please hit stop key !',0DH,0AH,0
C14E 61736520
C152 68697420
C156 73746F70
C15A 206B6579
C15E 20210D0A
C162 00
C163 73746F70 BREAK1:DEFB 'stop',0
C167 00
C168 20212107 BREAK2:DEFB ' !!',7,0
C16C 00
               ;
C16D                  END


*GC100
0 please hit stop key !
10 please hit stop key !
20 please hit stop key !
30 please hit stop key !
40 please hit stop key !
50 please hit stop key !
60 please hit stop key !
70 please hit stop key !
80 please hit stop key !
90 please hit stop key !
100 please hit stop key !
110 please hit stop key !
120 please hit stop key !
130 please hit stop key !
140 please hit stop key !
150 please hit stop key !
160 please hit stop key !
170 please hit stop key !
180 please hit stop key !
190 please hit stop key !
200 please hit stop key !
210 please hit stop key !
stop in 210 !!
*
```

2D13H　行番号の出力

アドレス　2D13H

機　能　行番号の出力

レジスタ　AFBCDEHL

解　説　レジスタHLに格納された16ビットのデータを，無符号の10進整数として，CRT画面上の現カーソル位置に表示します．

インタプリタでは，行番号の表示等に，このシステム・サブルーチンを使用しています．

また，EB49H番地の出力フラグが01H～7FHの場合には，プリンタへの出力を行います．

サンプル

```
                    ;
                    ; --- display numbers with two bytes ---
                    ;
                            ORG   0C100H
                    ;
C100 210000                 LD    HL,0
                    ;
C103 CD44C1         LOOP:   CALL  DSPHL              ; display hexa decimal
                    ;
C106 3E48                   LD    A,'H'
C108 CD5702                 CALL  0257H
C10B 3E20                   LD    A,' '
C10D CD5702                 CALL  0257H
C110 3E3D                   LD    A,'='
C112 CD5702                 CALL  0257H
C115 3E20                   LD    A,' '
C117 CD5702                 CALL  0257H
                    ;
C11A E5                     PUSH  HL
C11B CD132D                 CALL  2D13H              ;*display as unsigned decimal number
C11E E1                     POP   HL
                    ;
C11F 3E0D                   LD    A,0DH
C121 CD5702                 CALL  0257H
C124 3E0A                   LD    A,0AH
C126 CD5702                 CALL  0257H
                    ;
C129 CD3BC1                 CALL  WAIT               ; wait a moment
                    ;
C12C 23                     INC   HL                 ; increment the number
                    ;
C12D 7C                     LD    A,H                ;
C12E B5                     OR    L                  ; check end or not
C12F CA665C                 JP    Z,5C66H            ;
                    ;
C132 DB09                   IN    A,(09H)            ;
C134 CB47                   BIT   0,A                ; jump to system monitor if stop key
C136 CA665C                 JP    Z,5C66H            ;
                    ;
C139 18C8                   JR    LOOP               ; and repeat
                    ;
                    ; wait subroutine
```

```
                ;
C13B 01FFFF     WAIT:  LD   BC,0FFFFH
C13E 0B         WAIT1: DEC  BC
C13F 78                LD   A,B
C140 B1                OR   C
C141 20FB              JR   NZ,WAIT1
C143 C9                RET
                ;
                ; display hexa decimal number in register hl subroutine
                ;
                ;    inputs   : the number in register hl
                ;    destroys : register a,f,b
                ;
C144 7C         DSPHL: LD   A,H
C145 CD49C1            CALL DSPACC
C148 7D                LD   A,L
                ;
C149 47         DSPACC:LD   B,A
C14A 0F                RRCA
C14B 0F                RRCA
C14C 0F                RRCA
C14D 0F                RRCA
C14E CD52C1            CALL HALF
C151 78                LD   A,B
                ;
C152 E60F       HALF:  AND  0FH
C154 FE0A              CP   10
C156 3802              JR   C,NUMBER
C158 C607              ADD  A,7
                ;
C15A C630       NUMBER:ADD  A,30H
C15C CD5702            CALL 0257H            ; display a character
C15F C9                RET
                ;
C160                   END
```

```
*GC100
0000H = 0
0001H = 1
0002H = 2
0003H = 3
0004H = 4
0005H = 5
0006H = 6
0007H = 7
0008H = 8
0009H = 9
000AH = 10
000BH = 11
000CH = 12
000DH = 13
000EH = 14
000FH = 15
0010H = 16
0011H = 17
0012H = 18
0013H = 19
0014H = 20
0015H = 21

*
```

2D22H　数値データから文字列への変換

アドレス	2D22H
機　能	数値データから文字列への変換
レジスタ	AFBCDEHL

解　説　EF45H番地に格納された数値データの型を示す数値（02H，04H，または08H）により，フローティング・アキュムレータ（FACC）に格納されているバイナリ形式の数値データをキャラクタ・コードの文字列に変換して，F0B7H番地から始まる変換バッファに格納し，データの末尾に00H（データ・エンド・マーク）を付け加えます．

リターン時には，文字列の先頭アドレスがレジスタHLに入ります．

具体的にはアキュムレータに00Hを入れて（フリー・フォーマットを指定）2D23H番地にエントリします．

サンプル

```
                 ;
                 ; --- input and display a number free format ---
                 ;
                        ORG   0C100H
                 ;
C100 212AC1      START: LD    HL,PROMPT
C103 CD21C1             CALL  DSPMSG
                 ;
C106 CD7E1B             CALL  1B7EH           ; screen editor
C109 DA665C             JP    C,5C66H         ; jump to system monitor if stop key
                 ;
C10C 23                 INC   HL
C10D CDBE2B             CALL  2BBEH           ; convert into binary code
                 ;
C110 CD222D             CALL  2D22H           ;*convert facc into character code
                 ;
C113 E5                 PUSH  HL
C114 213BC1             LD    HL,ANSWER
C117 CD21C1             CALL  DSPMSG
C11A E1                 POP   HL
                 ;
C11B CD21C1             CALL  DSPMSG          ; display the number
                 ;
C11E C300C1             JP    START
                 ;
                 ; display message subroutine
                 ;
                 ;    destroys : register a,f,h,l
                 ;
C121 7E          DSPMSG:LD    A,(HL)
C122 A7                 AND   A
C123 C8                 RET   Z
C124 CD5702             CALL  0257H           ; display a character
C127 23                 INC   HL
C128 18F7               JR    DSPMSG
                 ;
                 ; message area
                 ;
C12A 0D0A0A61    PROMPT:DEFB 0DH,0AH,0AH,'any number ? ',0
C12E 6E79206E
```

```
C132 756D6265
C136 72203F20
C13A 00
C13B 74686520 ANSWER:DEFB 'the number : ',0
C13F 6E756D62
C143 6572203A
C147 2000
                 ;
C149                      END


*GC100


any number ? 5
the number :  5

any number ? 5.5
the number :  5.5

any number ? 5#
the number :  5

any number ? &ha
the number :  10

any number ? &o777
the number :  511

any number ? 1.12345
the number :  1.12345

any number ?

*
```

２Ｄ２３Ｈ　数値から有フォーマット文字列への変換

アドレス	２Ｄ２３Ｈ
機　能	数値から有フォーマット文字列への変換
レジスタ	ＡＦＢＣＤＥＨＬ

解　説　ＥＦ４５Ｈ番地に格納された数値データの型を示す数値（０２Ｈ，０４Ｈ，または０８Ｈ）により，フローティング・アキュムレータ（ＦＡＣＣ）に格納されているバイナリ形式の数値データをキャラクタ・コードの文字列に変換して，Ｆ０Ｂ７Ｈ番地から始まる変換バッファに格納し，データの末尾に００Ｈ（データ・エンド・マーク）を付け加えます．

リターン時には，文字列の先頭アドレスがレジスタＨＬに入ります．

このシステム・サブルーチンでは，文字列データの生成時に様々なフォーマットを指定することができるため，ＢＡＳＩＣにおけるＰＲＩＮＴ　ＵＳＩＮＧ，ＬＰＲＩＮＴ　ＵＳＩＮＧ，またＦＯＲＴＲＡＮにおけるＦＯＲＭＡＴ，などと同等の機能を比較的容易に実現することができます．

フォーマットの指定は，アキュムレータおよびレジスタＢ，レジスタＣにパラメータを与えて行いますが，以下にそれぞれのパラメータの意味を示します．

●アキュムレータ

第７ビット……フォーマットの指定を行うか否かを指定します．すなわち，このビットが０の場合には以下のフォーマット指定はすべて無意味なものとなります．

第６ビット……このビットを１にセットした場合，数値の整数部が３桁毎に“，”で区切られます．

第５ビット……このビットを１にセットした場合，数値領域の左側に空白部分ができたとき，そこを“＊”で埋めます．

第４ビット……このビットを１にセットした場合，数値の直前に“￥”を付け加えます．

第３ビット……このビットを１にセットした場合，数値が負のときの“－”だけでなく，数値が正のときの“＋”も付け加えます．

第２ビット……このビットを１にセットした場合，“－”または“＋”を数値の直後に付け加えます．

第０ビット……このビットを１にセットした場合，数値は指数形式で変換されます．

●レジスタB

小数点より左（整数部分）の桁数を指定します．指定した桁数より数値の桁数が小さい場合には，右づめになります．

●レジスタC

小数点のためのフィールド1桁分を含む，小数点より右（小数部分）の桁数を指定します．

指定した領域より数値の桁数が大きい場合，数値の直前に“%”が付け加えられます．数値の丸めが領域より大きくなる原因となった場合も，丸めた数値の前に“%”が付加されます．

サンプル

```
                ;
                ; --- print using test program ---
                ;
                        ORG   0C100H
                ;
                ; input a number section
                ;
C100 21D8C1     START:  LD    HL,PRMPT1
C103 CDCFC1             CALL  DSPMSG
C106 CD7E1B             CALL  1B7EH             ; screen editor
C109 DA665C             JP    C,5C66H           ; jump to system monitor if stop key
C10C 23                 INC   HL
C10D CDBE2B             CALL  2BBEH             ; convert into binary code
                ;
                ; print using section
                ;
C110 AF                 XOR   A                 ; clear accumulator
C111 5F                 LD    E,A               ; clear register e
                ;
C112 21F3C1             LD    HL,PRMPT7         ; using ?
C115 CDCFC1             CALL  DSPMSG
C118 CD750F             CALL  0F75H             ; input a character from keyboard
C11B CD5702             CALL  0257H             ; echo-back a character
C11E CBEF               SET   5,A               ; convert into small letter
C120 FE79               CP    'y'
C122 C2BDC1             JP    NZ,DSPLAY         ; display in     free format
C125 CBFB               SET   7,E               ; display in not free format
                ;
C127 210CC2     COMMA:  LD    HL,PRMPT6         ; commas ?
C12A CDCFC1             CALL  DSPMSG
C12D CD750F             CALL  0F75H             ; input a character from keyboard
C130 CD5702             CALL  0257H             ; echo-back a characer
C133 CBEF               SET   5,A               ; convert into small letter
C135 FE79               CP    'y'
C137 2002               JR    NZ,ASTER          ; not display commas
C139 CBF3               SET   6,E               ;     display commas
                ;
C13B 2126C2     ASTER:  LD    HL,PRMPT5         ; asterisk ?
C13E CDCFC1             CALL  DSPMSG
C141 CD750F             CALL  0F75H             ; input a character from keyboard
C144 CD5702             CALL  0257H             ; echo-back a character
C147 CBEF               SET   5,A               ; convert into small letter
C149 FE79               CP    'y'
C14B 2002               JR    NZ,YEN            ; not display asterisks
C14D CBEB               SET   5,E               ;     display asterisks
                ;
C14F 2140C2     YEN:    LD    HL,PRMPT4         ; yen mark ?
C152 CDCFC1             CALL  DSPMSG
C155 CD750F             CALL  0F75H             ; input a character from keyboard
C158 CD5702             CALL  0257H             ; echo-back a character
```

```
C15B CBEF            SET  5,A             ; convert into small letter
C15D FE79            CP   'y'
C15F 2002            JR   NZ,PLUS         ; not display yen mark
C161 CBE3            SET  4,E             ;     display yen mark
              ;
C163 215AC2   PLUS:  LD   HL,PRMPT3       ; plus mark ?
C166 CDCFC1          CALL DSPMSG
C169 CD750F          CALL 0F75H           ; input a character from keyboard
C16C CD5702          CALL 0257H           ; echo-back a character
C16F CBEF            SET  5,A             ; convert into small letter
C171 FE79            CP   'y'
C173 2002            JR   NZ,SIGN         ; not display plus mark
C175 CBDB            SET  3,E             ;     display plus mark
              ;
C177 2174C2   SIGN:  LD   HL,PRMPT2       ; sign ?
C17A CDCFC1          CALL DSPMSG
C17D CD750F          CALL 0F75H           ; input a character from keyboard
C180 CD5702          CALL 0257H           ; echo-back a character
C183 CBEF            SET  5,A             ; convert into small letter
C185 FE79            CP   'y'
C187 2002            JR   NZ,INDEX        ; not display sign after a number
C189 CBD3            SET  2,E             ;     display sign after a number
              ;
C18B 218EC2   INDEX: LD   HL,PRMPT0       ; index format ?
C18E CDCFC1          CALL DSPMSG
C191 CD750F          CALL 0F75H           ; input a character from keyboard
C194 CD5702          CALL 0257H           ; echo-back a character
C197 CBEF            SET  5,A             ; convert into small letter
C199 FE79            CP   'y'
C19B 2002            JR   NZ,LEFT         ; not  in index format
C19D CBC3            SET  0,E             ;      in index format
              ;
C19F 21A8C2   LEFT:  LD   HL,PRMPTL       ; left of point ?
C1A2 CDCFC1          CALL DSPMSG
C1A5 CD750F          CALL 0F75H           ; input a character from keyboard
C1A8 CD5702          CALL 0257H           ; echo-back a character
C1AB D630            SUB  '0'             ; convert a character into binary
C1AD 47              LD   B,A
              ;
C1AE 21C2C2   RIGHT: LD   HL,PRMPTR       ; right of point ?
C1B1 CDCFC1          CALL DSPMSG
C1B4 CD750F          CALL 0F75H           ; input a character from keyboard
C1B7 CD5702          CALL 0257H           ; echo-back a character
C1BA D630            SUB  '0'             ; convert a character into binary
C1BC 4F              LD   C,A
              ;
C1BD 7B       DSPLAY:LD   A,E             ; set the most inportant agument
C1BE CD232D          CALL 2D23H           ;*convert facc into character code
C1C1 E5              PUSH HL
C1C2 21DCC2          LD   HL,ANSWER
C1C5 CDCFC1          CALL DSPMSG
C1C8 E1              POP  HL
C1C9 CDCFC1          CALL DSPMSG          ; display the number
              ;
C1CC C300C1          JP   START
              ;
              ; display message subroutine
              ;
              ;    destroys : register a,f,h,l
              ;
C1CF 7E       DSPMSG:LD   A,(HL)
C1D0 A7              AND  A
C1D1 C8              RET  Z
C1D2 CD5702          CALL 0257H           ; display a character
C1D5 23              INC  HL
C1D6 18F7            JR   DSPMSG
              ;
              ; message area
              ;
C1D8 0D0A0A61 PRMPT1:DEFB 0DH,0AH,0AH,'any number           ? ',0
C1DC 6E79206E
C1E0 756D6265
C1E4 72202020
C1E8 20202020
```

```
C1EC 20202020
C1F0 3F2000
                ;
C1F3 0D757369 PRMPT7:DEFB 0DH,      'using (y)          ? ',0
C1F7 6E672028
C1FB 79292020
C1FF 20202020
C203 20202020
C207 20203F20
C20B 00
C20C 0D0A636F PRMPT6:DEFB 0DH,0AH,  'commas (y)         ? ',0
C210 6D6D6173
C214 20287929
C218 20202020
C21C 20202020
C220 2020203F
C224 2000
C226 0D0A6173 PRMPT5:DEFB 0DH,0AH,  'asterisks (y)      ? ',0
C22A 74657269
C22E 736B7320
C232 28792920
C236 20202020
C23A 2020203F
C23E 2000
C240 0D0A7965 PRMPT4:DEFB 0DH,0AH,  'yen mark (y)       ? ',0
C244 6E206D61
C248 726B2028
C24C 79292020
C250 20202020
C254 2020203F
C258 2000
C25A 0D0A706C PRMPT3:DEFB 0DH,0AH,  'plus mark (y)      ? ',0
C25E 7573206D
C262 61726B20
C266 28792920
C26A 20202020
C26E 2020203F
C272 2000
C274 0D0A7369 PRMPT2:DEFB 0DH,0AH,  'sign (y)           ? ',0
C278 676E2028
C27C 79292020
C280 20202020
C284 20202020
C288 2020203F
C28C 2000
C28E 0D0A696E PRMPT0:DEFB 0DH,0AH,  'index format (y)   ? ',0
C292 64657820
C296 666F726D
C29A 61742028
C29E 79292020
C2A2 2020203F
C2A6 2000
                ;
C2A8 0D0A6C65 PRMPTL:DEFB 0DH,0AH,  'left of point (0-9)  ? ',0
C2AC 6674206F
C2B0 6620706F
C2B4 696E7420
C2B8 28302D39
C2BC 2920203F
C2C0 2000
C2C2 0D0A7269 PRMPTR:DEFB 0DH,0AH,  'right of point (0-9) ? ',0
C2C6 67687420
C2CA 6F662070
C2CE 6F696E74
C2D2 2028302D
C2D6 3929203F
C2DA 2000
                ;
C2DC 0D0A7468 ANSWER:DEFB 0DH,0AH,  'the number           : ',0
C2E0 65206E75
C2E4 6D626572
C2E8 20202020
C2EC 20202020
C2F0 2020203A
```

```
C2F4 2000
                      ;
C2F6                           END


*GC100


any number               ? 1234.56789
using (y)                ? y
commas (y)               ? y
asterisks (y)            ? y
yen mark (y)             ? y
plus mark (y)            ? y
sign (y)                 ? y
index format (y)         ? n
left of point (0-9)      ? 9
right of point (0-9)     ? 3
the number               : ***¥1,234.57+

any number               ? 9876.54321
using (y)                ? y
commas (y)               ? n
asterisks (y)            ? n
yen mark (y)             ? n
plus mark (y)            ? n
sign (y)                 ? n
index format (y)         ? y
left of point (0-9)      ? 9
right of point (0-9)     ? 9
the number               :  98765432.10000000D-04

any number               ? 1234.5
using (y)                ? n
the number               :  1234.5

any number               ? 330044005500#
using (y)                ? y
commas (y)               ? y
asterisks (y)            ? y
yen mark (y)             ? y
plus mark (y)            ? y
sign (y)                 ? y
index format (y)         ? n
left of point (0-9)      ? 9
right of point (0-9)     ? 0
the number               : %¥330,044,005,500+

any number               ? z80
using (y)                ? n
the number               :  0

any number               ?

*
```

３０９ＦＨ　無符号整数データから文字列への変換

アドレス	３０９ＦＨ
機　　能	無符号整数データから文字列への変換
レジスタ	ＡＦＢＣ－－ＨＬ

解　　説　フローティング・アキュムレータに格納された整数型の数値データを，５桁の無符号１０進整数としてキャラクタ・コードの文字列に変換し，レジスタＨＬによって指定するアドレスから格納します．

ただし，ゼロ・サプレスは行わず，レジスタＣが１のときには“ ，”を付加し，レジスタＢが１のときには先頭に“ ．”を付加します．

サンプル

```
               ;
               ; --- display numbers with two bytes ---
               ;
                       ORG  0C100H
               ;
C100 0648              LD   B,72               ;
C102 0E19              LD   C,25               ; set 72 characters and 25 lines
C104 CD3A09            CALL 093AH              ;
               ;
C107 0606              LD   B,6
C109 217AC1    TITLE1:LD    HL,MESAGE
C10C CD71C1            CALL DSPMSG
C10F 10F8              DJNZ TITLE1
               ;
C111 0648              LD   B,72
C113 3E2D              LD   A,'-'
C115 CD5702    TITLE2:CALL  0257H
C118 10FB              DJNZ TITLE2
               ;
C11A 210000            LD   HL,0               ; initialize the number
               ;
C11D CD55C1    LOOP:   CALL DSPHL              ; display hexa decimal
               ;
C120 3E20              LD   A,' '
C122 CD5702            CALL 0257H
               ;
C125 E5                PUSH HL
               ;
C126 22A8F0            LD   (0F0A8H),HL        ;
C129 3E02              LD   A,2                ; set the number into facc
C12B 3245EF            LD   (0EF45H),A         ;
               ;
C12E 2100ED            LD   HL,0ED00H          ; use input buffer of basic
C131 010000            LD   BC,0000H
C134 CD9F30            CALL 309FH              ;*convert into decimal string
C137 2100ED            LD   HL,0ED00H          ; use input buffer of basic
C13A CD71C1            CALL DSPMSG
               ;
C13D E1                POP  HL
               ;
C13E 3E20              LD   A,' '
C140 CD5702            CALL 0257H
C143 CD5702            CALL 0257H
               ;
C146 23                INC  HL                 ; increment the number
               ;
```

```
C147 7C                 LD    A,H             ;
C148 B5                 OR    L               ; check end or not
C149 CA665C             JP    Z,5C66H         ;
                ;
C14C DB09               IN    A,(09H)         ;
C14E CB47               BIT   0,A             ; jump to system monitor if stop key
C150 CA665C             JP    Z,5C66H         ;
                ;
C153 18C8               JR    LOOP            ; jump to loop if not stop key
                ;
                ; display hexa decimal number in register hl subroutine
                ;
                ;    inputs   : number in register hl
                ;    destroys : register a,f,b
                ;
C155 7C         DSPHL:  LD    A,H
C156 CD5AC1             CALL  DSPACC
C159 7D                 LD    A,L
                ;
C15A 47         DSPACC:LD     B,A
C15B 0F                 RRCA
C15C 0F                 RRCA
C15D 0F                 RRCA
C15E 0F                 RRCA
C15F CD63C1             CALL  HALF
C162 78                 LD    A,B
                ;
C163 E60F       HALF:   AND   0FH
C165 FE0A               CP    10
C167 3802               JR    C,NUMBER
C169 C607               ADD   A,7
                ;
C16B C630       NUMBER:ADD    A,30H
C16D CD5702             CALL  0257H           ; display a character
C170 C9                 RET
                ;
                ; display message subroutine
                ;
C171 7E         DSPMSG:LD     A,(HL)
C172 A7                 AND   A
C173 C8                 RET   Z
C174 CD5702             CALL  0257H           ; display a character
C177 23                 INC   HL
C178 18F7               JR    DSPMSG
                ;
                ; message area
                ;
C17A 6865782E   MESAGE:DEFB 'hex.  dec.   ',0
C17E 20206465
C182 632E2020
C186 00
                ;
C187                    END
```

```
*GC100
 hex. dec.   hex. dec.   hex. dec.   hex. dec.   hex. dec.   hex. dec.
 ---------------------------------------------------------------------
 0000 00000  0001 00001  0002 00002  0003 00003  0004 00004  0005 00005
 0006 00006  0007 00007  0008 00008  0009 00009  000A 00010  000B 00011
 000C 00012  000D 00013  000E 00014  000F 00015  0010 00016  0011 00017
 0012 00018  0013 00019  0014 00020  0015 00021  0016 00022  0017 00023
 0018 00024  0019 00025  001A 00026  001B 00027  001C 00028  001D 00029
 001E 00030  001F 00031  0020 00032  0021 00033  0022 00034  0023 00035
 0024 00036  0025 00037  0026 00038  0027 00039  0028 00040  0029 00041
 002A 00042  002B 00043  002C 00044  002D 00045  002E 00046  002F 00047
 0030 00048  0031 00049  0032 00050  0033 00051  0034 00052  0035 00053
 0036 00054  0037 00055  0038 00056  0039 00057  003A 00058  003B 00059
 003C 00060  003D 00061  003E 00062  003F 00063  0040 00064  0041 00065
 0042 00066  0043 00067  0044 00068  0045 00069  0046 00070  0047 00071
 0048 00072  0049 00073  004A 00074  004B 00075  004C 00076  004D 00077
 004E 00078  004F 00079  0050 00080  0051 00081  0052 00082  0053 00083
 0054 00084  0055 00085  0056 00086  0057 00087  0058 00088  0059 00089
 005A 00090  005B 00091  005C 00092  005D 00093  005E 00094  005F 00095
 0060 00096  0061 00097  0062 00098  0063 00099  0064 00100  0065 00101
 0066 00102  0067 00103  0068 00104  0069 00105  006A 00106  006B 00107
 006C 00108  006D 00109  006E 00110  006F 00111  0070 00112  0071 00113
 0072 00114  0073 00115  0074 00116  0075 00117  0076 00118  0077 00119
 0078 00120  0079 00121  007A 00122
 *
```

31A1H　平方根の計算

アドレス	31A1H
機　　能	平方根の計算
レジスタ	AFBCDEHL

解　　説　フローティング・アキュムレータに格納された単精度型実数の，平方根の値を求め，同じくフローティング・アキュムレータに与えてもどります．

フローティング・アキュムレータに与えた入力パラメータが負数の場合には，“Illegal function call” エラーが発生しインタプリタの制御下に入ります．

サンプル

```
                ;
                ; --- sqr function test progtam ---
                ;
                        ORG   0C100H
                ;
C100 213AC1     START:  LD    HL,PROMPT
C103 CD31C1             CALL  DSPMSG
                ;
C106 CD7E1B             CALL  1B7EH           ; screen editor
C109 DA665C             JP    C,5C66H         ; system monitor if stop key
                ;
C10C 23                 INC   HL
C10D E5                 PUSH  HL
                ;
C10E CDBE2B             CALL  2BBEH           ; convert into binary code
C111 CDB327             CALL  27B3H           ; convert into single precision
C114 CDA131             CALL  31A1H           ;*sqr function
C117 CD222D             CALL  2D22H           ; convert into character code
                ;
C11A EB                 EX    DE,HL           ; top address of result in reg-de
                ;
C11B 2148C1             LD    HL,ANS1
C11E CD31C1             CALL  DSPMSG
                ;
C121 E1                 POP   HL              ; top address of a input number
C122 CD31C1             CALL  DSPMSG          ; display a input number again
                ;
C125 2154C1             LD    HL,ANS2
C128 CD31C1             CALL  DSPMSG
                ;
C12B EB                 EX    DE,HL           ; top address of result in reg-hl
C12C CD31C1             CALL  DSPMSG          ; display result
                ;
C12F 18CF               JR    START
                ;
                ; display message subroutine
                ;
C131 7E         DSPMSG:LD     A,(HL)
C132 A7                 AND   A
C133 C8                 RET   Z
C134 CD5702             CALL  0257H           ; display a character
C137 23                 INC   HL
C138 18F7               JR    DSPMSG
                ;
                ; message area
                ;
C13A 0D0A6120   PROMPT:DEFB 0DH,0AH,'a number ? ',0
C13E 6E756D62
C142 6572203F
```

```
C146 2000
C148 20202020 ANS1:  DEFB '      sqr ( ',0
C14C 20737172
C150 20282000
C154 20292069 ANS2:  DEFB ' ) is equal to ',0
C158 73206571
C15C 75616C20
C160 746F2000
              ;
C164                 END


*GC100

a number ? 0
     sqr ( 0 ) is equal to  0
a number ? 1
     sqr ( 1 ) is equal to  1
a number ? 2
     sqr ( 2 ) is equal to  1.41421
a number ? 4
     sqr ( 4 ) is equal to  2
a number ? 400
     sqr ( 400 ) is equal to  20
a number ? 1000000000000000
     sqr ( 1000000000000000 ) is equal to  3.16228E+07
a number ? 100000000
     sqr ( 100000000 ) is equal to  10000
a number ? 0.09
     sqr ( 0.09 ) is equal to  .3
a number ? abc
     sqr ( abc ) is equal to  0
a number ? -1
Illegal function call
Ok
```

３１Ｆ３Ｈ　eに対する指数関数の計算

アドレス　３１Ｆ３Ｈ

機　　能　eに対する指数関数の計算

レジスタ　ＡＦＢＣＤＥＨＬ

解　　説　フローティング・アキュムレータに格納された単精度型実数の，自然対数の底eに対する指数関数の値を求め，同じくフローティング・アキュムレータに与えてもどります．

フローティング・アキュムレータに与える入力パラメータが，８７．３３６５５より大きい場合には，“Overflow”エラーが発生します．

サンプル

```
               ;
               ; --- exp function test progtam ---
               ;
                       ORG  0C100H
               ;
C100 213AC1    START:  LD   HL,PROMPT
C103 CD31C1            CALL DSPMSG
               ;
C106 CD7E1B            CALL 1B7EH           ; screen editor
C109 DA665C            JP   C,5C66H         ; system monitor if stop key
               ;
C10C 23                INC  HL
C10D E5                PUSH HL
               ;
C10E CDBE2B            CALL 2BBEH           ; convert into binary code
C111 CDB327            CALL 27B3H           ; convert into single precision
C114 CDF331            CALL 31F3H           ;*exp function
C117 CD222D            CALL 2D22H           ; convert into character code
               ;
C11A EB                EX   DE,HL           ; top address of result in reg-de
               ;
C11B 2148C1            LD   HL,ANS1
C11E CD31C1            CALL DSPMSG
               ;
C121 E1                POP  HL              ; top address of a input number
C122 CD31C1            CALL DSPMSG          ; display a input number again
               ;
C125 2154C1            LD   HL,ANS2
C128 CD31C1            CALL DSPMSG
               ;
C12B EB                EX   DE,HL           ; top address of result in reg-hl
C12C CD31C1            CALL DSPMSG          ; display result
               ;
C12F 18CF              JR   START
               ;
               ; display message subroutine
               ;
C131 7E        DSPMSG:LD    A,(HL)
C132 A7                AND  A
C133 C8                RET  Z
C134 CD5702            CALL 0257H           ; display a character
C137 23                INC  HL
C138 18F7              JR   DSPMSG
               ;
               ; message area
               ;
```

```
C13A 0D0A6120 PROMPT:DEFB 0DH,0AH,'a number ? ',0
C13E 6E756D62
C142 6572203F
C146 2000
C148 20202020 ANS1:  DEFB '     exp ( ',0
C14C 20657870
C150 20282000
C154 20292069 ANS2:  DEFB ' ) is equal to ',0
C158 73206571
C15C 75616C20
C160 746F2000
              ;
C164                 END


*GC100

a number ? 0.001
     exp ( 0.001 ) is equal to  1.001
a number ? 1
     exp ( 1 ) is equal to  2.71828
a number ? 2
     exp ( 2 ) is equal to  7.38906
a number ? 3
     exp ( 3 ) is equal to  20.0855
a number ? 4
     exp ( 4 ) is equal to  54.5982
a number ? 5
     exp ( 5 ) is equal to  148.413
a number ? abc
     exp ( abc ) is equal to  1
a number ? 0
     exp ( 0 ) is equal to  1
a number ? 87
     exp ( 87 ) is equal to  6.07601E+37
a number ? 88
Overflow
Ok
```

3283H　乱数の発生

アドレス	3283H
機　能	乱数の発生
レジスタ	AFBCDEHL

解　説　0以上1未満の乱数を単精度型の実数として発生し，フローティング・アキュムレータに与えてもどります．

発生する乱数は，ＢＡＳＩＣのＲＵＮおよびＣＬＥＡＲが実行されるごとに同系列をとり，フローティング・アキュムレータに与える単精度実数型の入力パラメータによってつぎのように異なります．

　パラメータが負の場合，新しい乱数系列を設定します．

　パラメータが0の場合，1つ前に発生した乱数の値をとります．

　パラメータが正の場合，つぎの乱数を発生します．

サンプル

```
                ;
                ; --- generate and display random numbers ---
                ;
                        ORG   0C100H
                ;
C100 2131C1             LD    HL,TITLE
C103 CD28C1             CALL  DSPMSG
                ;
C106 0664               LD    B,100           ; appoint loop counter
                ;
C108 C5         LOOP:   PUSH  BC
                ;
C109 210000             LD    HL,0            ;
C10C 22A8F0             LD    (0F0A8H),HL     ;
C10F 2681               LD    H,81H           ; set single precision of 1
C111 22AAF0             LD    (0F0AAH),HL     ;   into floating accumulator
C114 3E04               LD    A,04            ;
C116 3245EF             LD    (0EF45H),A      ;
                ;
C119 CD8332             CALL  3283H           ;*rnd function
                ;
C11C CD222D             CALL  2D22H           ; convert into character code
C11F CD28C1             CALL  DSPMSG          ; display a random number
                ;
C122 C1                 POP   BC
C123 10E3               DJNZ  LOOP
                ;
C125 C3665C             JP    5C66H           ; jump to system monitor's entry
                ;
                ; display message subroutine
                ;
C128 7E         DSPMSG:LD     A,(HL)
C129 A7                 AND   A
C12A C8                 RET   Z
C12B CD5702             CALL  0257H           ; display a character
C12E 23                 INC   HL
C12F 18F7               JR    DSPMSG
                ;
                ; message area
                ;
```

```
C131 2D2D2D20 TITLE: DEFB '--- random numbers ---',0DH,0AH,0
C135 72616E64
C139 6F6D206E
C13D 756D6265
C141 7273202D
C145 2D2D0D0A
C149 00
                  ;
C14A                    END
```

```
*GC100
--- random numbers ---
 .585608 .208326 .748217 .863982 .30259
.968811 .259876 .630002 .507997 .23225 .
0612217 .744806 .628905 .138237 .0923359
 .521203 .294037 .496946 .205632 .515778
 .0356391 .982939 .719593 .705331 .60776
 .0206582 .449692 .0950858 .750213 .8021
33 .674913 .377068 .549213 .344769 .4641
12 .717874 .279036 .209502 .256304 .6890
23 .0764643 .600416 .824226 .845947 .224
968 .672405 .0823059 .809365 .168721 .73
5916 .422197 .741722 .558669 .1714 .6155
55 .793136 .625767 .749037 .811342 .6351
47 .332455 .538262 .42986 .929347 .53015
7 .69302 .173859 .659925 .781217 .131739
 .229435 .065044 .172119 .575302 .898291
 .0123984 .0722502 .507155 .817158 .8019
66 .0600475 .423976 .216693 .345024 .127
256 .302549 .576178 .410597 .941808 .594
264 .263846 .446635 .68406 .0927208 .157
349 .440018 .627279 .735533 .863298 .085
1214
*
```

3 2 F 6 H　余弦（コサイン）の計算

アドレス	3 2 F 6 H
機　能	余弦（コサイン）の計算
レジスタ	A F B C D E H L

解　説　フローティング・アキュムレータに格納された単精度型実数の単位をラジアン（π／１８０×角度）とした場合の三角関数コサイン（余弦）の値を求め，同じくフローティング・アキュムレータに与えてもどります.

サンプル

```
                ;
                ; --- cos function test progtam ---
                ;
                        ORG   0C100H
                ;
C100 213AC1     START: LD    HL,PROMPT
C103 CD31C1            CALL  DSPMSG
                ;
C106 CD7E1B            CALL  1B7EH            ; screen editor
C109 DA665C            JP    C,5C66H          ; system monitor if stop key
                ;
C10C 23                INC   HL
C10D E5                PUSH  HL
                ;
C10E CDBE2B            CALL  2BBEH            ; convert into binary code
C111 CDB327            CALL  27B3H            ; convert into single precision
C114 CDF632            CALL  32F6H            ;*cos function
C117 CD222D            CALL  2D22H            ; convert into character code
                ;
C11A EB                EX    DE,HL            ; top address of result in reg-de
                ;
C11B 2148C1            LD    HL,ANS1
C11E CD31C1            CALL  DSPMSG
                ;
C121 E1                POP   HL               ; top address of a input number
C122 CD31C1            CALL  DSPMSG           ; display a input number again
                ;
C125 2154C1            LD    HL,ANS2
C128 CD31C1            CALL  DSPMSG
                ;
C12B EB                EX    DE,HL            ; top address of result in reg-hl
C12C CD31C1            CALL  DSPMSG           ; display result
                ;
C12F 18CF              JR    START
                ;
                ; display message subroutine
                ;
C131 7E         DSPMSG:LD    A,(HL)
C132 A7                AND   A
C133 C8                RET   Z
C134 CD5702            CALL  0257H            ; display a character
C137 23                INC   HL
C138 18F7              JR    DSPMSG
                ;
                ; message area
                ;
C13A 0D0A6120   PROMPT:DEFB 0DH,0AH,'a number ? ',0
C13E 6E756D62
C142 6572203F
C146 2000
C148 20202020   ANS1:  DEFB '     cos ( ',0
```

```
C14C 20636F73
C150 20282000
C154 20292069 ANS2:  DEFB ´ ) is equal to ´,0
C158 73206571
C15C 75616C20
C160 746F2000
                 ;
C164                   END
```

```
*GC100

a number ? 1
     cos ( 1 ) is equal to  .540302
a number ? 1.1
     cos ( 1.1 ) is equal to  .453596
a number ? 1.2
     cos ( 1.2 ) is equal to  .362358
a number ? 1.3
     cos ( 1.3 ) is equal to  .267499
a number ? 1.4
     cos ( 1.4 ) is equal to  .169967
a number ? 1.5
     cos ( 1.5 ) is equal to  .0707371
a number ? 1.6
     cos ( 1.6 ) is equal to -.0291995
a number ?

*
Ok
print tab(29);cos(1.6)
                             -.0291995
Ok
```

32FCH　正弦（サイン）の計算

アドレス	32FCH
機　　能	正弦（サイン）の計算
レジスタ	AFBCDEHL

解　　説　フローティング・アキュムレータに格納された単精度型実数の単位をラジアン（π／180×角度）とした場合の三角関数サイン（正弦）の値を求め，同じくフローティング・アキュムレータに与えてもどります．

サンプル

```
                ;
                ; --- sin function test progtam ---
                ;
                        ORG   0C100H
                ;
C100 213AC1     START: LD    HL,PROMPT
C103 CD31C1            CALL  DSPMSG
                ;
C106 CD7E1B            CALL  1B7EH            ; screen editor
C109 DA665C            JP    C,5C66H          ; system monitor if stop key
                ;
C10C 23                INC   HL
C10D E5                PUSH  HL
                ;
C10E CDBE2B            CALL  2BBEH            ; convert into binary code
C111 CDB327            CALL  27B3H            ; convert into single precision
C114 CDF632            CALL  32F6H            ;*sin function
C117 CD222D            CALL  2D22H            ; convert into character code
                ;
C11A EB                EX    DE,HL            ; top address of result in reg-de
                ;
C11B 2148C1            LD    HL,ANS1
C11E CD31C1            CALL  DSPMSG
                ;
C121 E1                POP   HL               ; top address of a input number
C122 CD31C1            CALL  DSPMSG           ; display a input number again
                ;
C125 2154C1            LD    HL,ANS2
C128 CD31C1            CALL  DSPMSG
                ;
C12B EB                EX    DE,HL            ; top address of result in reg-hl
C12C CD31C1            CALL  DSPMSG           ; display result
                ;
C12F 18CF              JR    START
                ;
                ; display message subroutine
                ;
C131 7E         DSPMSG:LD    A,(HL)
C132 A7                AND   A
C133 C8                RET   Z
C134 CD5702            CALL  0257H            ; display a character
C137 23                INC   HL
C138 18F7              JR    DSPMSG
                ;
                ; message area
                ;
C13A 0D0A6120   PROMPT:DEFB 0DH,0AH,'a number ? ',0
C13E 6E756D62
C142 6572203F
C146 2000
C148 20202020   ANS1:  DEFB '     sin ( ',0
```

```
C14C 2073696E
C150 20282000
C154 20292069 ANS2:  DEFB ´ ) is equal to ´,0
C158 73206571
C15C 75616C20
C160 746F2000
                 ;
C164                      END
```

```
*GC100

a number ? 0
     sin ( 0 ) is equal to  1
a number ? 0.1
     sin ( 0.1 ) is equal to  .995004
a number ? 0.2
     sin ( 0.2 ) is equal to  .980067
a number ? 0.3
     sin ( 0.3 ) is equal to  .955337
a number ? 0.4
     sin ( 0.4 ) is equal to  .921061
a number ? 5
     sin ( 5 ) is equal to  .283662
a number ? 6
     sin ( 6 ) is equal to  .96017
a number ? 7
     sin ( 7 ) is equal to  .753903
a number ? 8
     sin ( 8 ) is equal to -.1455
a number ? 9
     sin ( 9 ) is equal to -.91113
a number ?

*
```

３３５ＤＨ　正接値（タンジェント）の計算

アドレス	３３５ＤＨ
機　能	正接値（タンジェント）の計算
レジスタ	ＡＦＢＣＤＥＨＬ

解　説　フローティング・アキュムレータに格納された単精度型実数の単位をラジアン（π／１８０×角度）とした場合の三角関数タンジェント（正接値）の値を求め，同じくフローティング・アキュムレータに与えてもどります．

ただし，正接値（タンジェント）の値は計算式『tan θ=sin θ/cos θ』に準じて算出するため，入力パラメータに対する余弦（コサイン）の値が０（ゼロ）をとる場合（π／２，３π／２，５π／２等の場合），“Division by zero” エラーを発生しインタプリタの制御下に入ります．

サンプル

```
                ;
                ; --- tan function test progtam ---
                ;
                        ORG  0C100H
                ;
C100 213AC1     START: LD   HL,PROMPT
C103 CD31C1            CALL DSPMSG
                ;
C106 CD7E1B            CALL 1B7EH           ; screen editor
C109 DA665C            JP   C,5C66H         ; system monitor if stop key
                ;
C10C 23                INC  HL
C10D E5                PUSH HL
                ;
C10E CDBE2B            CALL 2BBEH           ; convert into binary code
C111 CDB327            CALL 27B3H           ; convert into single precision
C114 CD5D33            CALL 335DH           ;*tan function
C117 CD222D            CALL 2D22H           ; convert into character code
                ;
C11A EB                EX   DE,HL           ; top address of result in reg-de
                ;
C11B 2148C1            LD   HL,ANS1
C11E CD31C1            CALL DSPMSG
                ;
C121 E1                POP  HL              ; top address of a input number
C122 CD31C1            CALL DSPMSG          ; display a input number again
                ;
C125 2154C1            LD   HL,ANS2
C128 CD31C1            CALL DSPMSG
                ;
C12B EB                EX   DE,HL           ; top address of result in reg-hl
C12C CD31C1            CALL DSPMSG          ; display result
                ;
C12F 18CF              JR   START
                ;
                ; display message subroutine
                ;
C131 7E         DSPMSG:LD   A,(HL)
C132 A7                AND  A
C133 C8                RET  Z
C134 CD5702            CALL 0257H           ; display a character
C137 23                INC  HL
```

```
C138 18F7             JR    DSPMSG
                ;
                ; message area
                ;
C13A 0D0A6120 PROMPT:DEFB 0DH,0AH,'a number ? ',0
C13E 6E756D62
C142 6572203F
C146 2000
C148 20202020 ANS1:  DEFB '      tan ( ',0
C14C 2074616E
C150 20282000
C154 20292069 ANS2:  DEFB ' ) is equal to ',0
C158 73206571
C15C 75616C20
C160 746F2000
                ;
C164                  END


*GC100

a number ? 0.001
     tan ( 0.001 ) is equal to  1.00003E-03
a number ? 0.01
     tan ( 0.01 ) is equal to  .0100003
a number ? 0.1
     tan ( 0.1 ) is equal to  .100335
a number ? 1
     tan ( 1 ) is equal to  1.55741
a number ? 10
     tan ( 10 ) is equal to  .64836
a number ? 100
     tan ( 100 ) is equal to -.587219
a number ? 1000
     tan ( 1000 ) is equal to  1.47022
a number ? 10000
     tan ( 10000 ) is equal to  .320945
a number ? 100000
     tan ( 100000 ) is equal to -.0368322
a number ?

*
```

３３７２Ｈ　逆正接（アーク・タンジェント）の計算

アドレス	３３７２Ｈ
機　　能	逆正接（アーク・タンジェント）の計算
レジスタ	ＡＦＢＣＤＥＨＬ

解　　説　フローティング・アキュムレータに格納された単精度型実数の単位をラジアン（π／１８０×角度）とした場合の三角関数アーク・タンジェント(逆正接）の値を求め，同じくフローティング・アキュムレータに与えてもどります．

演算結果は，－π／２からπ／２までの値になります．

サンプル

```
              ;
              ; --- atn function test progtam ---
              ;
                      ORG   0C100H
              ;
C100 213AC1   START:  LD    HL,PROMPT
C103 CD31C1           CALL  DSPMSG
              ;
C106 CD7E1B           CALL  1B7EH           ; screen editor
C109 DA665C           JP    C,5C66H         ; system monitor if stop key
              ;
C10C 23               INC   HL
C10D E5               PUSH  HL
              ;
C10E CDBE2B           CALL  2BBEH           ; convert into binary code
C111 CDB327           CALL  27B3H           ; convert into single precision
C114 CD7233           CALL  3372H           ;*atn function
C117 CD222D           CALL  2D22H           ; convert into character code
              ;
C11A EB               EX    DE,HL           ; top address of result in reg-de
              ;
C11B 2148C1           LD    HL,ANS1
C11E CD31C1           CALL  DSPMSG
              ;
C121 E1               POP   HL              ; top address of a input number
C122 CD31C1           CALL  DSPMSG          ; display a input number again
              ;
C125 2154C1           LD    HL,ANS2
C128 CD31C1           CALL  DSPMSG
              ;
C12B EB               EX    DE,HL           ; top address of result in reg-hl
C12C CD31C1           CALL  DSPMSG          ; display result
              ;
C12F 18CF             JR    START
              ;
              ; display message subroutine
              ;
C131 7E       DSPMSG:LD     A,(HL)
C132 A7               AND   A
C133 C8               RET   Z
C134 CD5702           CALL  0257H           ; display a character
C137 23               INC   HL
C138 18F7             JR    DSPMSG
              ;
              ; message area
              ;
C13A 0D0A6120 PROMPT:DEFB 0DH,0AH,'a number ? ',0
C13E 6E756D62
C142 6572203F
```

```
C146 2000
C148 20202020 ANS1:  DEFB '      atn ( ',0
C14C 2061746E
C150 20282000
C154 20292069 ANS2:  DEFB ' ) is equal to ',0
C158 73206571
C15C 75616C20
C160 746F2000
               ;
C164                 END


*GC100

a number ? 1
     atn ( 1 ) is equal to  .785398
a number ? 2
     atn ( 2 ) is equal to  1.10715
a number ? 3
     atn ( 3 ) is equal to  1.24905
a number ? 4
     atn ( 4 ) is equal to  1.32582
a number ? 5
     atn ( 5 ) is equal to  1.3734
a number ? 6
     atn ( 6 ) is equal to  1.40565
a number ? 10000000000
     atn ( 10000000000 ) is equal to  1.5708
a number ?

*
print tab(37);atn(10000000000)
                                    1.5708
Ok
```

3BF9H　エラー出力

アドレス　3BF9H

機　　能　エラー出力

レジスタ　AFBCDEHL

解　　説　レジスタEに格納されたエラー・コード（1～255）に対応するエラーを出力して，インタプリタの制御下に入ります．

ただし，レジスタEに格納されたエラー・コードに対応するエラーが存在しない場合には，“Unprintable error　” エラーを出力します．

また，レジスタEに00H（ゼロ）が与えられた場合には，無限ループに入り込んでしまうため注意が必要です．

サンプル

```
                ;
                ; --- output error of basic ---
                ;
                        ORG   0C100H
                ;
                ; input error code section
                ;
C100 213BC1             LD    HL,PROMPT        ; error code ?
C103 CD32C1             CALL  DSPMSG
                ;
C106 1E00               LD    E,0              ; initialize input error code
C108 3E30               LD    A,'0'            ; initialize input character
                ;
C10A F5         INLOOP:PUSH  AF                ;
C10B 7B                 LD    A,E              ;
C10C 87                 ADD   A,A              ;
C10D 87                 ADD   A,A              ;
C10E 83                 ADD   A,E              ;
C10F 87                 ADD   A,A              ; e <-- e * 10 + input number
C110 5F                 LD    E,A              ;
C111 F1                 POP   AF               ;
C112 D630               SUB   '0'              ;
C114 83                 ADD   A,E              ;
C115 5F                 LD    E,A              ;
                ;
C116 CD750F             CALL  0F75H            ; input a character from keyboard
C119 CD5702             CALL  0257H            ; echo-back a character
                ;
C11C FE03               CP    03H
C11E CA665C             JP    Z,5C66H          ; jump to system monitor if stop key
                ;
C121 FE0D               CP    0DH
C123 20E5               JR    NZ,INLOOP
                ;
C125 3E0A               LD    A,0AH            ; line feed code in accumulator
C127 CD5702             CALL  0257H            ; display a character
                ;
C12A 1C                 INC   E                ;
C12B 1D                 DEC   E                ; system monitor if error code is ze
C12C CA665C             JP    Z,5C66H          ;
                ;
                ; output error of basic section
                ;
C12F C3F93B             JP    3BF9H            ;*output error and jump to basic
```

```
                ;
                ; display message subroutine
                ;
                ;    inputs   : top address of message in register hl
                ;    destroys : register a,f,h,l
                ;
C132 7E         DSPMSG:LD   A,(HL)
C133 A7                AND  A
C134 C8                RET  Z
C135 CD5702            CALL 0257H            ; display a character
C138 23                INC  HL
C139 18F7              JR   DSPMSG
                ;
                ; message area
                ;
C13B 6572726F PROMPT:DEFB 'error code ? ',0
C13F 7220636F
C143 6465203F
C147 2000
                ;
C149                   END


*GC100
error code ? 1
NEXT without FOR
Ok
mon
*GC100
error code ? 2
Syntax error
Ok
mon
*GC100
error code ? 3
RETURN without GOSUB
Ok
mon
*GC100
error code ? 4
Out of DATA
Ok
mon
*GC100
error code ? 5
Illegal function call
Ok
mon
*GC100
error code ? 10
Redimensioned array
Ok
mon
*GC100
error code ? 11
Division by zero
Ok
mon
*GC100
error code ? 12
Illegal direct
Ok
mon
*GC100
error code ? 20
RESUME without error
Ok
mon
*GC100
error code ? 30
Unprintable error
Ok
```

3D76H　BASICテキストのリンク1

アドレス	3D76H
機　能	BASIテキストのリンク1
レジスタ	AFBCDEHL

解　説　EB54H～EB55H番地に格納されているプログラム・エリアの先頭アドレスから，レジスタHLをポインタとしてプログラム・エリアを上位に向かってサーチして行き，つぎの行の先頭アドレスを見つけたら現在行のリンク・ポインタに代入する作業を，次行の先頭2バイトが共に00Hになるまで続けます．

終了時には，プログラムのエンド・アドレスがレジスタHLに，レジスタHLの値から1を減じたものがレジスタDEに，それぞれ格納されます．

サンプル

```
                ;
                ; --- search text and set link-pointers ---
                ;
                       ORG  0C100H
                ;
C100 2146C1            LD   HL,MSG1
C103 CD3DC1            CALL DSPMSG
                ;
C106 2A54EB            LD   HL,(0EB54H)
C109 CD21C1            CALL DSPHL
                ;
C10C 2164C1            LD   HL,MSG2
C10F CD3DC1            CALL DSPMSG
                ;
C112 CD763D            CALL 3D76H              ;*seach text and set link-pointers
C115 CD21C1            CALL DSPHL
                ;
C118 216AC1            LD   HL,MSG3
C11B CD3DC1            CALL DSPMSG
                ;
C11E C3665C            JP   5C66H
                ;
                ; display hexa decimal number in register hl subroutine
                ;
                ;    inputs   : number in register hl
                ;    destroys : register a,f,b
                ;
C121 7C         DSPHL: LD   A,H
C122 CD26C1            CALL DSPACC
C125 7D                LD   A,L
                ;
C126 47         DSPACC:LD   B,A
C127 0F                RRCA
C128 0F                RRCA
C129 0F                RRCA
C12A 0F                RRCA
C12B CD2FC1            CALL HALF
C12E 78                LD   A,B
                ;
C12F E60F       HALF:  AND  0FH
C131 FE0A              CP   10
C133 3802              JR   C,NUMBER
```

```
C135 C607            ADD  A,7
                ;
C137 C630       NUMBER:ADD  A,30H
C139 CD5702          CALL 0257H           ; display a character
C13C C9              RET
                ;
                ; display message subroutine
                ;
                ;    inputs   : top address of message in register hl
                ;    destroys : register a,f,h,l
                ;
C13D 7E         DSPMSG:LD   A,(HL)
C13E A7              AND  A
C13F C8              RET  Z
C140 CD5702          CALL 0257H           ; display a character
C143 23              INC  HL
C144 18F7            JR   DSPMSG
                ;
                ; massage area
                ;
C146 6C696E6B MSG1:  DEFB 'linked !',0DH,0AH,'basic text on from ',0
C14A 65642021
C14E 0D0A6261
C152 73696320
C156 74657874
C15A 206F6E20
C15E 66726F6D
C162 2000
C164 6820746F MSG2:  DEFB 'h to ',0
C168 2000
C16A 68202121 MSG3:  DEFB 'h !!',0
C16E 00
                ;
C16F                 END
```

```
new
Ok
mon
*GC100
linked !
basic text on from 8021h to 8022h !!
*D8021,8022
8021 00 00
*
Ok
100 print "This is the text !!"
run
This is the text !!
Ok
mon
*GC100
linked !
basic text on from 8021h to 803Eh !!
*D8021,803E
8021 3D 80 64 00 91 20 22 54 68 69 73 20 69 73 20
8030 74 68 65 20 74 65 78 74 20 21 21 22 00 00 00
*
```

3D79H　BASICテキストのリンク2

アドレス	3D79H
機　　能	BASICテキストのリンク2
レジスタ	AFBCDEHL

解　　説　レジスタＨＬによって指定するアドレスから，レジスタＨＬをポインタとしてプログラム・エリアを上位に向かってサーチして行き，つぎの行の先頭アドレスを見つけたら現在行のリンク・ポインタに代入する作業を，次行の先頭２バイトが共に００Ｈになるまで続けます．

終了時には，プログラムのエンド・アドレスがレジスタＨＬに，レジスタＨＬの値から１を減じたものがレジスタＤＥに，それぞれ格納されます．

3D7AH　BASICテキストのリンク3

アドレス	3D7AH
機　　能	BASICテキストのリンク3
レジスタ	AFBCDEHL

解　　説　レジスタＤＥによって指定するアドレスから，レジスタＨＬをポインタとしてプログラム・エリアを上位に向かってサーチして行き，つぎの行の先頭アドレスを見つけたら現在行のリンク・ポインタに代入する作業を，次行の先頭２バイトが共に００Ｈになるまで続けます．

終了時には，プログラムのエンド・アドレスがレジスタＨＬに，レジスタＨＬの値から１を減じたものがレジスタＤＥに，それぞれ格納されます．

3DC1H　BASICの行番号サーチ

アドレス	3DC1H
機　能	BASICの行番号サーチ
レジスタ	AFBCDEHL

解　説　ＢＡＳＩＣのテキスト・エリア中から，レジスタＤＥによって行番号を指定するＢＡＳＩＣプログラムの格納されているアドレスをさがし出すためのシステム・サブルーチンです．

さがし出したアドレスは，レジスタＢＣおよびレジスタＨＬに格納しますが，ゼロ・フラグ（Ｚ）およびキャリ・フラグ（ＣＹ）の変化によって，つぎに示す３通りの実行結果に分類することができます．

ゼロ・フラグ（Ｚ），キャリ・フラグ（ＣＹ）共にセットされた場合……指定した行番号に相当するＢＡＳＩＣのプログラム行が存在することを示します．レジスタＢＣには指定プログラム行の先頭アドレス（リンク・ポインタを含む）が，レジスタＨＬには指定したつぎのプログラム行の先頭アドレスが，それぞれ格納されます．

ゼロ・フラグ（Ｚ）がセットされ，キャリ・フラグ（ＣＹ）がリセットされた場合……指定したプログラム行がＢＡＳＩＣのテキスト中に存在せずに，かつ指定した行番号より大きい行番号のプログラム行が１行も存在しないことを示します．レジスタＢＣおよびレジスタＨＬには，ＢＡＳＩＣのテキスト・エンド・マークを示す連続した２バイトの００Ｈが格納されている最初のアドレスが格納されます．

ゼロ・フラグ（Ｚ），キャリ・フラグ（ＣＹ）共にリセットされた場合……指定したプログラム行がＢＡＳＩＣのテキスト中に存在しないことを示しますが，指定した行番号より大きい行番号のプログラム行は存在することを示します．レジスタＢＣには指定した行番号より大きい行番号を持つ最初のプログラム行の先頭アドレスが，レジスタＨＬにはそのつぎのプログラム行の先頭アドレスが，それぞれ格納されます．

サンプル

```
;
; --- search the basic text for the line ---
;
         ORG   0C100H
;
; input line number section
```

```
                ;
C100 2166C1             LD   HL,PROMPT
C103 CD5DC1             CALL DSPMSG
                ;
C106 210000             LD   HL,0
C109 3E30               LD   A,'0'
                ;
C10B E5         INLOOP:PUSH HL
C10C D1                 POP  DE
C10D 29                 ADD  HL,HL
C10E 29                 ADD  HL,HL
C10F 19                 ADD  HL,DE
C110 29                 ADD  HL,HL
C111 D630               SUB  '0'
C113 5F                 LD   E,A
C114 1600               LD   D,0
C116 19                 ADD  HL,DE
                ;
C117 CD750F             CALL 0F75H            ; input a character from keyboard
C11A CD5702             CALL 0257H            ; echo-back a character
C11D FE0D               CP   0DH
C11F 20EA               JR   NZ,INLOOP
                ;
C121 EB                 EX   DE,HL            ; set line number in register de
                ;
                ; search for the line section
                ;
C122 CDC13D             CALL 3DC1H            ;*search the text for the line
                ;
                ; display result section
                ;
C125 3011               JR   NC,ERROR         ; not found if reset carry flag
                ;
C127 2191C1             LD   HL,FOUND         ; line from
C12A CD5DC1             CALL DSPMSG
                ;
C12D CD41C1             CALL DSPBC            ; display address
C130 3E48               LD   A,'H'
C132 CD5702             CALL 0257H            ; display a character
C135 C3665C             JP   5C66H            ; jump to system monitor
                ;
C138 2175C1     ERROR: LD   HL,NFOUND         ; line number not found !!
C13B CD5DC1             CALL DSPMSG
C13E C3665C             JP   5C66H            ; jump to system monitor
                ;
                ; display register bc hexa decimal subroutine
                ;
                ;    inputs   : number in register bc
                ;    destroys : reguster a,f,b
                ;
C141 78         DSPBC: LD   A,B
C142 CD46C1             CALL DSPACC
C145 79                 LD   A,C
                ;
C146 47         DSPACC:LD   B,A
C147 0F                 RRCA
C148 0F                 RRCA
C149 0F                 RRCA
C14A 0F                 RRCA
C14B CD4FC1             CALL HALF
C14E 78                 LD   A,B
                ;
C14F E60F       HALF:  AND  0FH
C151 FE0A               CP   10
C153 3802               JR   C,NUMBER
C155 C607               ADD  A,7
                ;
C157 C630       NUMBER:ADD  A,30H
C159 CD5702             CALL 0257H            ; display a character
C15C C9                 RET
                ;
                ; display message subroutine
                ;
                ;    inputs   : top address of message in register hl
```

```
                ;    destroys : register a,f,h,l
                ;
C15D 7E         DSPMSG:LD   A,(HL)
C15E A7                AND  A
C15F C8                RET  Z
C160 CD5702            CALL 0257H             ; display a character
C163 23                INC  HL
C164 18F7              JR   DSPMSG
                ;
                ; message area
                ;
C166 6C696E65 PROMPT:DEFB 'line number ? ',0
C16A 206E756D
C16E 62657220
C172 3F2000
C175 0D0A6C69 NFOUND:DEFB 0DH,0AH,'line number not found !!',07H,0
C179 6E65206E
C17D 756D6265
C181 72206E6F
C185 7420666F
C189 756E6420
C18D 21210700
C191 0D0A6C69 FOUND: DEFB 0DH,0AH,'line from ',0
C195 6E652066
C199 726F6D20
C19D 00
                ;
C19E                   END
```

```
list
100 PRINT "line number 100"
110 PRINT "line number 110"
120 PRINT "line number 120"
130 PRINT "line number 130"
140 PRINT "line number 140"
150 PRINT "line number 150"
Ok
mon
*GC100
line number ? 10
line number not found !!
*GC100
line number ? 100
line from 8021H
*GC100
line number ? 110
line from 8039H
*GC100
line number ? 120
line from 8051H
*GC100
line number ? 130
line from 8069H
*GC100
line number ? 135
line number not found !!
*GC100
line number ? 60000
line number not found !!
*D8000,80FF
8000 E6 F1 E9 F1 C9 00 C9 00 7F 23 A9 23 C9 00 C9 00
8010 C9 00 C9 00 C9 00 C9 00 C9 00 C9 00 C9 00 C9 00
8020 00 39 80 64 00 91 20 22 6C 69 6E 65 20 6E 75 6D
8030 62 65 72 20 31 30 30 22 00 51 80 6E 00 91 20 22
8040 6C 69 6E 65 20 6E 75 6D 62 65 72 20 31 31 30 22
8050 00 69 80 78 00 91 20 22 6C 69 6E 65 20 6E 75 6D
8060 62 65 72 20 31 32 30 22 00 81 80 82 00 91 20 22
8070 6C 69 6E 65 20 6E 75 6D 62 65 72 20 31 33 30 22
8080 00 99 80 8C 00 91 20 22 6C 69 6E 65 20 6E 75 6D
8090 62 65 72 20 31 34 30 22 00 B1 80 96 00 91 20 22
80A0 6C 69 6E 65 20 6E 75 6D 62 65 72 20 31 35 30 22
80B0 00 00 00 96 00 00 00 00 00 00 00 00 00 00 00 00
```

3E5CH　インプット

アドレス	3E5CH
機　能	インプット
レジスタ	AFBCDEHL

解　説　テキスト・スクリーン上の現カーソル位置に入力を促すプロンプトと１個のスペース“？　”を表示後，キーボードから１ライン分のデータを入力し，ＢＡＳＩＣのインプット・バッファ（ＥＣ９６Ｈ～ＥＤ９５Ｈ番地）に格納してもどります．ただし，入力データは２５４バイトまでが有効で，２５４バイトを越えて入力した場合には，先頭からの２５４バイトが入力データと見なされます．

データの入力は，ＲＥＴＵＲＮキー（ＣＴＲＬ＋Ｍ）またはＳＴＯＰキー（ＣＴＲＬ＋Ｃ）の入力によって終了しますが，前者の場合にはキャリ・フラグ（ＣＹ)を０にリセットし，後者の入力によって中断した場合にはキャリ・フラグ(ＣＹ）を１にセットしてもどります．

このサブルーチンからのリターン時には，レジスタＨＬにＥＣ９５Ｈ（インプット・バッファの開始アドレス―１）が，アキュムレータには最後に入力されたキャラクタのキャラクタ・コード（０３Ｈまたは０ＤＨ）が与えられます．

サンプル

```
                ;
                ; --- my personal history statement ---
                ;
                        ORG   0C100H
                ;
C100 21B1C1             LD    HL,NAME
C103 CDA8C1             CALL  DSPMSG
C106 CD5C3E             CALL  3E5CH
C109 DA665C             JP    C,5C66H
C10C 23                 INC   HL
C10D 111FC2             LD    DE,BNAME
C110 011400             LD    BC,20
C113 EDB0               LDIR
                ;
C115 21BAC1             LD    HL,ADRS
C118 CDA8C1             CALL  DSPMSG
C11B CD5C3E             CALL  3E5CH
C11E DA665C             JP    C,5C66H
C121 23                 INC   HL
C122 1133C2             LD    DE,BADRS
C125 016400             LD    BC,100
C128 EDB0               LDIR
                ;
C12A 21C3C1             LD    HL,PHONE
C12D CDA8C1             CALL  DSPMSG
C130 CD5C3E             CALL  3E5CH
C133 DA665C             JP    C,5C66H
C136 23                 INC   HL
C137 1197C2             LD    DE,BPHONE
```

```
C13A 011400            LD   BC,20
C13D EDB0              LDIR
                ;
C13F 21CCC1            LD   HL,SEX
C142 CDA8C1            CALL DSPMSG
C145 CD5C3E            CALL 3E5CH
C148 DA665C            JP   C,5C66H
C14B 23                INC  HL
C14C 11ABC2            LD   DE,BSEX
C14F 011400            LD   BC,20
C152 EDB0              LDIR
                ;
C154 21D5C1            LD   HL,AGE
C157 CDA8C1            CALL DSPMSG
C15A CD5C3E            CALL 3E5CH
C15D DA665C            JP   C,5C66H
C160 23                INC  HL
C161 11BFC2            LD   DE,BAGE
C164 011400            LD   BC,20
C167 EDB0              LDIR
                ;
C169 21DEC1            LD   HL,ANAME
C16C CDA8C1            CALL DSPMSG
C16F 211FC2            LD   HL,BNAME
C172 CDA8C1            CALL DSPMSG
                ;
C175 21EBC1            LD   HL,AADRS
C178 CDA8C1            CALL DSPMSG
C17B 2133C2            LD   HL,BADRS
C17E CDA8C1            CALL DSPMSG
                ;
C181 21F8C1            LD   HL,APHONE
C184 CDA8C1            CALL DSPMSG
C187 2197C2            LD   HL,BPHONE
C18A CDA8C1            CALL DSPMSG
                ;
C18D 2105C2            LD   HL,ASEX
C190 CDA8C1            CALL DSPMSG
C193 21ABC2            LD   HL,BSEX
C196 CDA8C1            CALL DSPMSG
                ;
C199 2112C2            LD   HL,AAGE
C19C CDA8C1            CALL DSPMSG
C19F 21BFC2            LD   HL,BAGE
C1A2 CDA8C1            CALL DSPMSG
                ;
C1A5 C3665C            JP   5C66H
                ;
                ; display message subroutine
                ;
                ;    inputs   : top address of message in register hl
                ;    destroys : register a,f,h,l
                ;
C1A8 7E         DSPMSG:LD   A,(HL)
C1A9 A7                AND  A
C1AA C8                RET  Z
C1AB CD5702            CALL 0257H            ; display a character
C1AE 23                INC  HL
C1AF 18F7              JR   DSPMSG
                ;
                ; message area
                ;
C1B1 6E616D65 NAME:  DEFB 'name     ',0
C1B5 20202020
C1B9 00
C1BA 61646472 ADRS:  DEFB 'address  ',0
C1BE 65737320
C1C2 00
C1C3 70686F6E PHONE: DEFB 'phone    ',0
C1C7 65202020
C1CB 00
C1CC 73657820 SEX:   DEFB 'sex      ',0
C1D0 20202020
C1D4 00
```

```
C1D5 61676520 AGE:   DEFB 'age      ',0
C1D9 20202020
C1DD 00
              ;
C1DE 0D0A6E61 ANAME: DEFB 0DH,0AH,'name    : ',0
C1E2 6D652020
C1E6 20203A20
C1EA 00
C1EB 0D0A6164 AADRS: DEFB 0DH,0AH,'address : ',0
C1EF 64726573
C1F3 73203A20
C1F7 00
C1F8 0D0A7068 APHONE:DEFB 0DH,0AH,'phone   : ',0
C1FC 6F6E6520
C200 20203A20
C204 00
C205 0D0A7365 ASEX:  DEFB 0DH,0AH,'sex     : ',0
C209 78202020
C20D 20203A20
C211 00
C212 0D0A6167 AAGE:  DEFB 0DH,0AH,'age     : ',0
C216 65202020
C21A 20203A20
C21E 00
              ;
              ; data buffer
              ;
C21F          BNAME: DEFS 20
C233          BADRS: DEFS 100
C297          BPHONE:DEFS 20
C2AB          BSEX:  DEFS 20
C2BF          BAGE:  DEFS 20
              ;
C2D3                 END


*GC100
name    ? KIYOSHI KAWAMURA
address ? 12-22-516 NISHI-MACHI XXXXX-KU YOKOHAMA 235
phone   ? 045-XXX-6237
sex     ? A MALE
age     ? 2X

name    : KIYOSHI KAWAMURA
address : 12-22-516 NISHI-MACHI XXXXX-KU YOKOHAMA 235
phone   : 045-XXX-6237
sex     : A MALE
age     : 2X
*
```

４０９５Ｈ　レジスタ・ペアの比較

アドレス	４０９５Ｈ
機　能	レジスタ・ペアの比較
レジスタ	ＡＦ－－－－－－

解　説　レジスタＨＬに格納された１６ビットの無符号整数と，レジスタＤＥに格納された１６ビットの無符号整数を比較し，結果をゼロ・フラグ（Ｚ）およびキャリ・フラグ（ＣＹ）にセットしてもどります．

それぞれのフラグは，レジスタＨＬとレジスタＤＥのデータが等しい場合にはゼロ・フラグ（Ｚ）を１にセットし，前者の方が小さい場合にはキャリ・フラグ（ＣＹ）を１にセットします．

サンプル

```
                 ;
                 ; --- dump memory with ascii character ---
                 ;
                        ORG  0C100H
                 ;
C100 110000             LD   DE,0000H          ; top address for dump memory
C103 21FFFF             LD   HL,0FFFFH         ; end address for dump memory
                 ;
                 ; display address section
                 ;
C106 7A          LOOPL: LD   A,D
C107 CD44C1             CALL DSPACC
C10A 7B                 LD   A,E
C10B CD44C1             CALL DSPACC
C10E 3E3A               LD   A,':'
C110 DF                 RST  18H
                 ;
                 ; dump hexa decimal section
                 ;
C111 D5                 PUSH DE
C112 0610               LD   B,16
C114 3E20        LOOPS1:LD   A,' '
C116 DF                 RST  18H
C117 1A                 LD   A,(DE)
C118 CD44C1             CALL DSPACC            ; dump a memory
C11B 13                 INC  DE
C11C 10F6               DJNZ LOOPS1
C11E D1                 POP  DE
                 ;
C11F 3E20               LD   A,' '             ;
C121 DF                 RST  18H               ; display three spaces
C122 DF                 RST  18H               ;
                 ;
                 ; dump ascii character section
                 ;
C123 0610               LD   B,16
C125 1A          LOOPS2:LD   A,(DE)
C126 FE20               CP   20H
C128 3002               JR   NC,ASCII
C12A 3E2E               LD   A,'.'
C12C DF          ASCII: RST  18H               ; dump ascii character
C12D 13                 INC  DE
C12E 10F5               DJNZ LOOPS2
                 ;
```

```
C130 3E0D           LD    A,0DH
C132 DF             RST   18H
C133 3E0A           LD    A,0AH
C135 DF             RST   18H
               ;
C136 CDF10C         CALL  0CF1H             ; check stop key
C139 DA665C         JP    C,5C66H           ; jump to system monitor if stop key
               ;
C13C CD9540         CALL  4095H             ;*compare register hl-register de
C13F 30C5           JR    NC,LOOPL
               ;
C141 C3665C         JP    5C66H             ; jump to system monitor if end
               ;
               ; output hexa decimal number in accumulator subroutine
               ;
               ;    inputs   : binary code in accumulator
               ;    destroys : register a,f
               ;
C144 F5        DSPACC:PUSH AF
C145 0F             RRCA
C146 0F             RRCA
C147 0F             RRCA
C148 0F             RRCA
C149 CD4DC1         CALL  HALF
C14C F1             POP   AF
               ;
C14D E60F      HALF:  AND   0FH
C14F FE0A           CP    10
C151 3802           JR    C,NUMBER
C153 C607           ADD   A,7
               ;
C155 C630      NUMBER:ADD   A,30H
C157 DF             RST   18H
C158 C9             RET
               ;
C159                END


*GC100
0000: F3 31 FF FF C3 3B 00 00 C3 6A 00 C3 57 17 AB F0  円1  テ;..テj.テW.ォ×
0010: C3 59 42 C3 6A 00 DA 0C C3 A6 40 F3 0B C3 7E 50  テYBテj.レ.テヲ@円.テ~P
0020: C3 DA F1 C3 9C 27 88 0C C3 DD F1 C3 60 0D 46 0C  テレ円テ,'|.テン円テ`.F.
0030: C3 E0 F1 9F 0F C3 57 02 C3 E3 F1 AF 32 75 EA CD  テ=円ノ.テW.テ╡円ッ2u◆ヘ
0040: F1 0C 30 C7 18 CD 00 00 00 00 00 00 00 00 00 00  円.0ヌ.ヘ..........
0050: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0060: 00 00 00 00 00 00 00 00 00 00 F3 CD 4C 16 AF 32  ..........円ヘL.ッ2
0070: 60 EA 3D 32 61 EA 01 81 00 C5 01 00 FF C5 C3 F6  `◆=2a◆.▁.ナ.. ナテ合
0080: 08 CD B0 F1 3A 55 EB ED 47 3E FF D3 E4 32 55 EA  .ヘー円:U♠○G> モ◢2U◆
0090: ED 5E FB 01 81 3C C3 10 3C C5 D5 F5 CD B3 F1 32  ○^ .▁<テ.<ナユ時ヘウ円2
00A0: B9 ED 3E 01 38 02 3E 02 CD 21 01 F1 F5 47 38 0F  ケ○>.8.>.ヘ!.円時G8.
00B0: 28 10 CD 5A 01 E6 40 CA 68 01 CD 39 01 18 03 CD  (.ヘZ.◥@ハh.ヘ9...ヘ
00C0: 4C 01 CD 5A 01 F1 D1 C1 B7 C9 C9 DB 40 E6 08 EE  L.ヘZ.円ムチキノノロ@◥./
00D0: 08 C9 C5 D5 F5 3A 67 EA E6 FE D3 40 F6 01 D3 40  .ノナユ時:g◆◥ モ@合.モ@
00E0: 32 67 EA CD 23 44 28 11 3E 91 CD 29 02 AF 32 C5  2g◆ヘ#D(.>┬ヘ).ッ2ナ
00F0: ED 3C 32 CB ED CD FF 00 A7 C1 F5 20 C5 18 C6 3E  ○<2ヒ○ヘ .ァチ時 ナ.ニ>
0100: 00 CD 7C 01 3E 05 32 CB ED 3E 07 CD 7C 01 CD E9  .ヘ¦.>.2ヒ○>.ヘ¦.ヘ♥
0110: 01 47 3E 10 0E 00 57 A0 28 01 0C 7A 07 30 F7 79  .G>...W (..z.0秒y
0120: C9 CD 7C 01 3A B9 ED CD 83 01 3A C6 ED CD 83 01  ノヘ¦.:ケ○ヘ▃.:ニ○ヘ▃.
0130: 78 CD 83 01 79 CD 83 01 C9 3E 03 CD 7C 01 0E 00  xヘ▃.yヘ▃.ノ>.ヘ¦...
0140: CD E9 01 12 13 0D 20 F8 05 20 F3 C9 0E 00 1A 13  ヘ♥....  . 円ノ....
0150: CD 83 01 0D 20 F8 05 20 F3 C9 3E 06 CD 7C 01 CD  ヘ▃..  . 円ノ>.ヘ¦.ヘ

*
```

４０Ａ６Ｈ　デバイスへの１バイト出力

アドレス	４０Ａ６Ｈ
機　能	デバイスへの１バイト出力
レジスタ	ーーーーーーーーー

解　説　ＥＢ４９Ｈ番地に格納されているデータが００Ｈの場合にはＣＲＴスクリーンへ，データが０１Ｈ～７ＦＨの場合にはプリンタへ，データが８０Ｈ～ＦＦＨの場合にはＣＭＴインタフェースへ，それぞれアキュムレータのデータ１バイトを出力します．

通常は，ＥＢ４９Ｈ番地に００Ｈを与え，ＣＲＴスクリーンへの１文字出力ルーチンとして使用します．

サンプル

```
                ;
                ; --- display or print out characters ---
                ;
                        ORG   0C100H
                ;
C100 AF         START:  XOR   A
C101 3249EB             LD    (0EB49H),A       ; select crt for output
                ;
C104 2164C1             LD    HL,PROMPT
C107 CD5BC1             CALL  DSPMSG
C10A CD750F             CALL  0F75H            ; input a character from keyboard
C10D FE03               CP    03H
C10F CA665C             JP    Z,5C66H          ; system monitor if stop key
                ;
C112 F5                 PUSH  AF               ;
C113 CD5702             CALL  0257H            ;
C116 3E0D               LD    A,0DH            ;
C118 CD5702             CALL  0257H            ; echo-back and line feed on crt
C11B 3E0A               LD    A,0AH            ;
C11D CD5702             CALL  0257H            ;
C120 F1                 POP   AF               ;
                ;
C121 FE63               CP    'c'
C123 2829               JR    Z,DSPLAY
C125 FE43               CP    'C'
C127 2825               JR    Z,DSPLAY
C129 FE64               CP    'd'
C12B 2821               JR    Z,DSPLAY
C12D FE44               CP    'D'
C12F 281D               JR    Z,DSPLAY
                ;
C131 FE6C               CP    'l'
C133 280E               JR    Z,LPRINT
C135 FE4C               CP    'L'
C137 280A               JR    Z,LPRINT
C139 FE70               CP    'p'
C13B 2806               JR    Z,LPRINT
C13D FE50               CP    'P'
C13F 2802               JR    Z,LPRINT
                ;
C141 18BD               JR    START
                ;
C143 217BC1     LPRINT:LD     HL,CARE
C146 CD5BC1             CALL  DSPMSG
```

```
                ;
C149 3E01               LD    A,1
C14B 3249EB             LD    (0EB49H),A        ; select printer for output
                ;
                ; output characters section
                ;
C14E 3E20       DSPLAY:LD    A,20H
                ;
C150 CDA640     LOOP:   CALL  40A6H             ;*output to each device
C153 3C                 INC   A
C154 FE00               CP    0FFH+1
C156 20F8               JR    NZ,LOOP
                ;
C158 C3665C             JP    5C66H             ; jump to system monitor
                ;
                ; display message subroutine
                ;
C15B 7E         DSPMSG:LD    A,(HL)
C15C A7                 AND   A
C15D C8                 RET   Z
C15E CD5702             CALL  0257H             ; display a character
C161 23                 INC   HL
C162 18F7               JR    DSPMSG
                ;
                ; message area
                ;
C164 63727420 PROMPT:DEFB ´crt or line-printer ? ´,0
C168 6F72206C
C16C 696E652D
C170 7072696E
C174 74657220
C178 3F2000
C17B 6C696E65 CARE:   DEFB ´line-printer active !!´,0AH,0DH,0
C17F 2D707269
C183 6E746572
C187 20616374
C18B 69766520
C18F 21210A0D
C193 00
                ;
C194                    END


*GC100
crt or line-printer ? l
line-printer active !!

*GC100
crt or line-printer ? d
 !"#$%&´()*+,-./0123456789:;<=>?@ABCDEFG
HIJKLMNOPQRSTUVWXYZ[¥]^_`abcdefghijklmno
pqrstuvwxyz{¦}~ ▁▂▃▄▅▆▇█▏▎▍▌▋▊▉┼┴┬┤├▔─│▕
┌┐└┘╭╮╰╯。「」、・ヲァィゥェォャュョッーアイウエオカキクケコサシスセソ
タチツテトナニヌネノハヒフヘホマミムメモヤユヨラリルレロワン゛゜═╞╪╡◢◣
♠♥♦♣●○╱╲╳円年月日時分秒
*
```

４ＢＥＡＨ　整数の除算２

アドレス	４ＢＥＡＨ
機　能	整数の除算２
レジスタ	ＡＦＢＣＤＥＨＬ

解　説　レジスタＤＥに格納された有符号１６ビットの整数を被除数とし，レジスタＨＬに格納された有符号１６ビットの整数を除数として，両者の間で除算を行い，演算結果をフローティング・アキュムレータに格納してもどります．

ただし，レジスタＨＬに与える除数が，００００Ｈ（ゼロ）であった場合にのみ，"Division by zero" エラーが発生しインタプリタの制御下に入ります．

このシステム・サブルーチンが２９５０Ｈ番地の整数除算サブルーチンと異なる点は，整数型で与えられた被除数および除数を単精度実数型に変換した上で，演算を行うことです．このためフローティング・アキュムレータに格納される演算結果は，単精度実数型として小数点以下まで正常に計算された値です．

サンプル

```
                 ;
                 ; --- division for integer ii ---
                 ;
                        ORG   0C100H
                 ;
C100 2148C1      START: LD    HL,PRMPT1
C103 CD3FC1             CALL  DSPMSG
C106 CD7E1B             CALL  1B7EH          ; screen editor
C109 DA665C             JP    C,5C66H        ; system monitor if stop key
C10C 23                 INC   HL
C10D CDBE2B             CALL  2BBEH          ; convert into binary code
C110 CD7F27             CALL  277FH          ; convert into integer
C113 ED5BA8F0           LD    DE,(0F0A8H)    ; store 1st argument into reg-de
C117 D5                 PUSH  DE
                 ;
C118 2158C1             LD    HL,PRMPT2
C11B CD3FC1             CALL  DSPMSG
C11E CD7E1B             CALL  1B7EH          ; screen editor
C121 23                 INC   HL
C122 CDBE2B             CALL  2BBEH          ; convert into binary code
C125 CD7F27             CALL  277FH          ; convert into integer
C128 2AA8F0             LD    HL,(0F0A8H)    ; set 2nd argument into reg-hl
                 ;
C12B D1                 POP   DE             ; set 1st argument into reg-de
                 ;
C12C CDEA4B             CALL  4BEAH          ;*division for integer ii
                 ;
C12F CD222D             CALL  2D22H          ; convert into character code
C132 E5                 PUSH  HL
C133 2165C1             LD    HL,ANSWER
C136 CD3FC1             CALL  DSPMSG
C139 E1                 POP   HL
C13A CD3FC1             CALL  DSPMSG         ; display result
                 ;
C13D 18C1               JR    START
                 ;
                 ; display message subroutine
                 ;
```

```
                ;     inputs   : top address of message in register hl
                ;     destroys : register a,f,h,l
                ;
C13F 7E         DSPMSG:LD   A,(HL)
C140 A7                AND  A
C141 C8                RET  Z
C142 CD5702            CALL 0257H           ; display a character
C145 23                INC  HL
C146 18F7              JR   DSPMSG
                ;
                ; message area
                ;
C148 0D0A0A20 PRMPT1:DEFB 0DH,0AH,0AH,'     x%    ? ',0
C14C 20202020
C150 78252020
C154 203F2000
C158 20202020 PRMPT2:DEFB '     y%    ? ',0
C15C 20792520
C160 20203F20
C164 00
C165 7825202F ANSWER:DEFB 'x% / y% --> ',0
C169 20792520
C16D 2D2D3E20
C171 00
                ;
C172                   END


*GC100


     x%    ? 20
     y%    ? 6
x% / y% -->  3.33333

     x%    ? 23.456
     y%    ? 10
x% / y% -->  2.3

     x%    ? -500
     y%    ? 20
x% / y% --> -25

     x%    ? 100
     y%    ? 1
x% / y% -->  100

     x%    ? 100
     y%    ? 0.5
Division by zero
Ok
```

４ＣＣＢＨ　英小文字から英大文字への変換１

|アドレス|　４ＣＣＢＨ
|機　　能|　英小文字から英大文字への変換１
|レジスタ|　ＡＦ－－－－－－

|解　　説|　レジスタＨＬの内容で指定するアドレスに格納されたデータを，アキュムレータに移動し，英小文字に相当するキャラクタ・コード（６１Ｈ～７ＡＨ）であれば英大文字（４１Ｈ～５ＡＨ）に変換します．

４ＣＣＣＨ　英小文字から英大文字への変換２

|アドレス|　４ＣＣＣＨ
|機　　能|　英小文字から英大文字への変換２
|レジスタ|　ＡＦ－－－－－－

|解　　説|　アキュムレータに格納されたデータが英小文字に相当するキャラクタ・コード（６１Ｈ～７ＡＨ）であれば英大文字（４１Ｈ～５ＡＨ）に変換します．

サンプル

```
                ;
                ; --- display capital characters ---
                ;
                        ORG   0C100H
                ;
                ; display normal characters section
                ;
C100 3E20               LD    A,20H

C102 CD5702     LOOP:   CALL  0257H              ; display a character
                ;
C105 3C                 INC   A
C106 FE00               CP    0FFH+1
C108 20F8               JR    NZ,LOOP
                ;
C10A 3E0D               LD    A,0DH
C10C CD5702             CALL  0257H
C10F 3E0A               LD    A,0AH
C111 CD5702             CALL  0257H
                ;
                ; display not small characters section
                ;
C114 3E20               LD    A,20H
                ;
C116 F5         LOOPC:  PUSH  AF
                ;
C117 CDCC4C             CALL  4CCCH              ;*convert small letter into capital
                ;
C11A CD5702             CALL  0257H              ; display a character
                ;
C11D F1                 POP   AF
                ;
C11E 3C                 INC   A
C11F FE00               CP    0FFH+1
C121 20F3               JR    NZ,LOOPC
                ;
C123 C3665C             JP    5C66H              ; jump to system monitor
                ;
C126                    END
```

```
*GC100
 !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFG
HIJKLMNOPQRSTUVWXYZ[¥]^_`abcdefghijklmno
pqrstuvwxyz{¦}~ ▁▂▃▄▅▆▇█▏▎▍▌▋▊▉┼┴┬┤├▔─▏▕
┌┐└┘╭╮╰╯ ｡｢｣､･ｦｧｨｩｪｫｬｭｮｯｰｱｲｳｴｵｶｷｸｹｺｻｼｽｾｿ
ﾀﾁﾂﾃﾄﾅﾆﾇﾈﾉﾊﾋﾌﾍﾎﾏﾐﾑﾒﾓﾔﾕﾖﾗﾘﾙﾚﾛﾜﾝﾞﾟ═╪╡▲▼
♠♥♦♣●○╱╲╳円年月日時分秒
 !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFG
HIJKLMNOPQRSTUVWXYZ[¥]^_`ABCDEFGHIJKLMNO
PQRSTUVWXYZ{¦}~ ▁▂▃▄▅▆▇█▏▎▍▌▋▊▉┼┴┬┤├▔─▏▕
┌┐└┘╭╮╰╯ ｡｢｣､･ｦｧｨｩｪｫｬｭｮｯｰｱｲｳｴｵｶｷｸｹｺｻｼｽｾｿ
ﾀﾁﾂﾃﾄﾅﾆﾇﾈﾉﾊﾋﾌﾍﾎﾏﾐﾑﾒﾓﾔﾕﾖﾗﾘﾙﾚﾛﾜﾝﾞﾟ═╪╡▲▼
♠♥♦♣●○╱╲╳円年月日時分秒
*
```

４ＣＥ９Ｈ　１６進文字列から数値データへの変換

アドレス	４ＣＥ９Ｈ
機　能	１６進文字列から数値データへの変換
レジスタ	ＡＦＢＣＤＥＨＬ

解　説　レジスタＨＬで指定するつぎのアドレスから１６進関係以外のコードが格納されているアドレスまでの１６進キャラクタ・コードによる文字列データを，バイナリ・コードに変換し，すでにレジスタＤＥに格納されているバイナリ・コードをシフトしながら加えて行き，結果をレジスタＤＥおよび整数型のフローティング・アキュムレータ（Ｆ０Ａ８Ｈ～Ｆ０Ａ９Ｈ番地）に格納してもどります．

ただし，１６進キャラクタ・コードによる文字列が5桁以上の場合には，“Overflow”エラーを発生してインタプリタの制御下に入ります．

サンプル

```
                ;
                ; --- hexa decimal vs decimal ---
                ;
                        ORG   0C100H
                ;
C100 114CC1     START:  LD    DE,PROMPT
C103 CD43C1             CALL  DSPMSG
                ;
C106 2195EC             LD    HL,0EC95H       ; top address-1 of input buffer
C109 23         INLOOP: INC   HL              ; increment pointer
C10A CD750F             CALL  0F75H           ; input a character from keyboard
C10D CD5702             CALL  0257H           ; echo-back it
C110 77                 LD    (HL),A          ; store it
C111 FE03               CP    03H
C113 CA665C             JP    Z,5C66H
C116 FE0D               CP    0DH
C118 20EF               JR    NZ,INLOOP
                ;
C11A 3600               LD    (HL),0          ; write end mark
                ;
C11C 2195EC             LD    HL,0EC95H       ; top address-1 of input buffer
C11F 110000             LD    DE,0            ; clear argumenter
C122 CDE94C             CALL  4CE9H           ;*convert into binary code
C125 EB                 EX    DE,HL           ; move result into register hl
                ;
C126 1169C1             LD    DE,UNSIGN
C129 CD43C1             CALL  DSPMSG
                ;
C12C E5                 PUSH  HL
C12D CD132D             CALL  2D13H           ; display unsigned decimal number
C130 E1                 POP   HL
                ;
C131 1183C1             LD    DE,SIGN
C134 CD43C1             CALL  DSPMSG
                ;
C137 CD9C27             CALL  279CH           ; store register hl into facc
C13A CD222D             CALL  2D22H           ; convert into character code
C13D EB                 EX    DE,HL
C13E CD43C1             CALL  DSPMSG          ; display result
                ;
```

```
C141 18BD           JR    START
               ;
               ; display message subroutine
               ;
               ;    inputs   : top address of message in register de
               ;    destroys : a,f,d,e
               ;
C143 1A        DSPMSG:LD    A,(DE)
C144 A7               AND   A
C145 C8               RET   Z
C146 CD5702           CALL  0257H          ; display a character
C149 13               INC   DE
C14A 18F7             JR    DSPMSG
               ;
               ; message area
               ;
C14C 0D0A0A32 PROMPT:DEFB 0DH,0AH,0AH,'2 bytes hexa decimal ? &h',0
C150 20627974
C154 65732068
C158 65786120
C15C 64656369
C160 6D616C20
C164 3F202668
C168 00
C169 0D0A2020 UNSIGN:DEFB 0DH,0AH,'    unsigned decimal : ',0
C16D 2020756E
C171 7369676E
C175 65642064
C179 6563696D
C17D 616C203A
C181 2000
C183 0D0A2020 SIGN:  DEFB 0DH,0AH,'      signed decimal :',0
C187 20202020
C18B 7369676E
C18F 65642064
C193 6563696D
C197 616C203A
C19B 00
               ;
C19C                  END


*GC100


2 bytes hexa decimal ? &h0
    unsigned decimal : 0
      signed decimal : 0

2 bytes hexa decimal ? &h7fff
    unsigned decimal : 32767
      signed decimal : 32767

2 bytes hexa decimal ? &h8000
    unsigned decimal : 32768
      signed decimal :-32768

2 bytes hexa decimal ? &h8001
    unsigned decimal : 32769
      signed decimal :-32767

2 bytes hexa decimal ? &hffff
    unsigned decimal : 65535
      signed decimal :-1

2 bytes hexa decimal ? &h
*
```

５２ＥＣＨ　文字列の出力１

アドレス	５２ＥＣＨ
機　能	文字列の出力１
レジスタ	ＡＦＢＣＤＥＨＬ

解　説　レジスタＨＬで指定するつぎのアドレスから００Ｈ（エンド・マーク）が格納されているアドレスまでに置かれた２５５バイト以内のデータを，ＥＢ４９Ｈ番地の出力フラグによって指定する出力デバイスへ出力します．

ＥＢ４９Ｈ番地に格納されている出力フラグの内容が，００Ｈの場合にはＣＲＴスクリーンへの出力，０１Ｈ～７ＦＨの場合にはプリンタへの出力，８０Ｈ～ＦＦＨの場合にはＣＭＴへの出力となりますが，通常はＥＢ４９Ｈ番地に００Ｈを与えてＣＲＴテキスト・スクリーンへのメッセージ表示サブルーチンとして使用します．

５２ＥＤＨ　文字列の出力２

アドレス	５２ＥＤＨ
機　能	文字列の出力２
レジスタ	ＡＦＢＣＤＥＨＬ

解　説　レジスタＨＬで指定するアドレスから００Ｈ（エンド・マーク）が格納されているアドレスまでに置かれた２５５バイト以内のデータを，ＥＢ４９Ｈ番地の出力フラグによって指定する出力デバイスへ出力します．

ＥＢ４９Ｈ番地に格納されている出力フラグの内容が，００Ｈの場合にはＣＲＴスクリーンへの出力，０１Ｈ～７ＦＨの場合にはプリンタへの出力，８０Ｈ～ＦＦＨの場合にはＣＭＴへの出力となりますが，通常はＥＢ４９Ｈ番地に００Ｈを与えてＣＲＴテキスト・スクリーンへのメッセージ表示サブルーチンとして使用します．

サンプル

```
                  ;
                  ; --- postscript ---
                  ;
                        ORG  0C100H
                  ;
C100 2109C1       START: LD   HL,SCRIPT
C103 CDED52             CALL 52EDH            ;*just display postscript
C106 C3665C             JP   5C66H
                  ;
                  ; postscript's area
                  ;
C109 504F5354 SCRIPT:DEFB 'POSTSCRIPT',0DH,0AH
C10D 53435249
C111 50540D0A
C115 46756E63         DEFB 'Functions of the NEC PC-8001 M'
C119 74696F6E
C11D 73206F66
C121 20746865
C125 204E4543
C129 2050432D
C12D 38303031
C131 204D
C133 41524B20         DEFB 'ARK TWO are diverse.',0DH,0AH
C137 54574F20
C13B 61726520
C13F 64697665
C143 7273652E
C147 0D0A
C149 49742063         DEFB 'It can be used for management,
C14D 616E2062
C151 65207573
C155 65642066
C159 6F72206D
C15D 616E6167
C161 656D656E
C165 742C
C167 206D6561         DEFB ' measurement, education, or si'
C16B 73757265
C16F 6D656E74
C173 2C206564
C177 75636174
C17B 696F6E2C
C17F 206F7220
C183 7369
C185 6D706C79         DEFB 'mply as a hobby.',0DH,0AH
C189 20617320
C18D 6120686F
C191 6262792E
C195 0D0A
C197 4920686F         DEFB 'I hope this book will serve as'
C19B 70652074
C19F 68697320
C1A3 626F6F6B
C1A7 2077696C
C1AB 6C207365
C1AF 72766520
C1B3 6173
C1B5 20616964         DEFB ' aid to a more advanced use of'
C1B9 20746F20
C1BD 61206D6F
C1C1 72652061
C1C5 6476616E
C1C9 63656420
C1CD 75736520
C1D1 6F66
C1D3 2069742E         DEFB ' it...',0
C1D7 2E2E00
                  ;
C1DA                    END
```

```
*GC100
POSTSCRIPT
Functions of the NEC PC-8001 MARK TWO are diverse.
It can be used for management, measurement, education, or simply as a hobby.
I hope this book will serve as aid to a more advanced use of it...
*
```